Data_scaling_task__Theodore Nguyen

July 13, 2022

[ ]:
```python
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
```

1 Our Dataset: Social_Network_Ads.csv

[ ]:
```python
sna = pd.read_csv('Social_Network_Ads.csv')
sna
```

[ ]:
```
      User ID  Gender  Age  EstimatedSalary  Purchased
0    15624510    Male   19            19000          0
1    15810944    Male   35            20000          0
2    15668575  Female   26            43000          0
3    15603246  Female   27            57000          0
4    15804002    Male   19            76000          0
..        …       …   …              …        …
395  15691863  Female   46            41000          1
396  15706071    Male   51            23000          1
397  15654296  Female   50            20000          1
398  15755018    Male   36            33000          0
399  15594041  Female   49            36000          1

[400 rows x 5 columns]
```

2 Trimmed Data Set

[ ]:
```python
# Trim data to only include columns: Age and EstimatedSalary
df = sna.drop(['User ID', 'Gender', 'Purchased'], axis = 1, inplace = False)
df
```

[ ]:
```
     Age  EstimatedSalary
0     19            19000
1     35            20000
2     26            43000
3     27            57000
4     19            76000
```

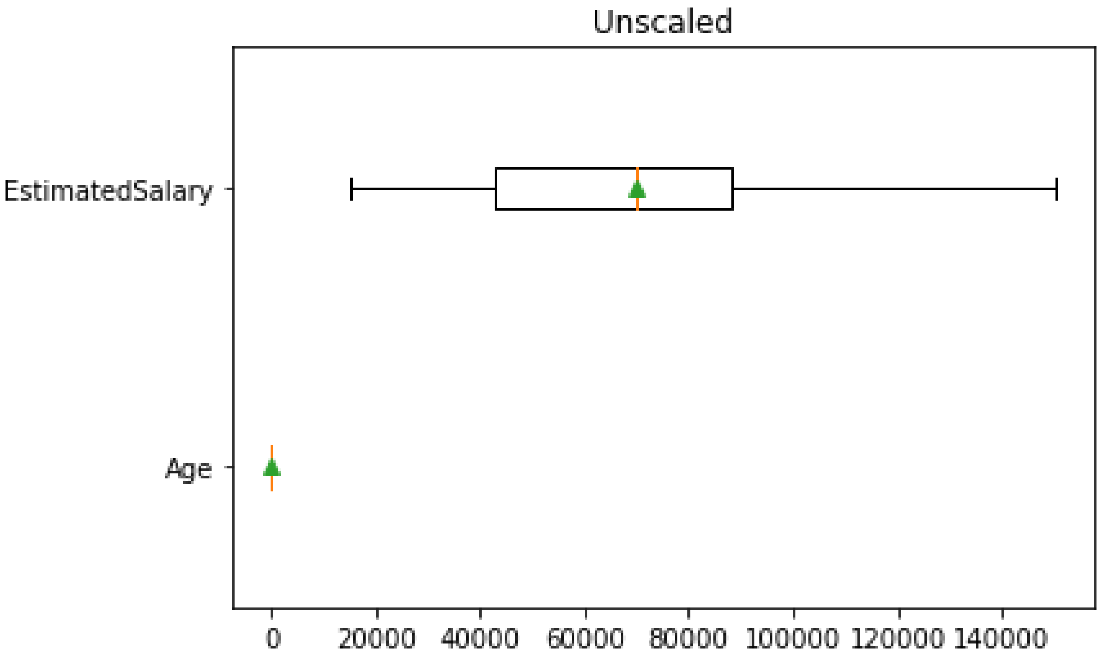
```
..   …             …
395  46            41000
396  51            23000
397  50            20000
398  36            33000
399  49            36000

[400 rows x 2 columns]
```

```
[ ]: columnLabels = ['Age', 'EstimatedSalary']
```

3 Before Scaling

```
[ ]: fig_Unscaled, ax = plt.subplots(1, figsize = (6, 4))
     ax.boxplot(df, vert = False, showmeans = True, labels = columnLabels)
     ax.set(title = 'Unscaled')
```

```
[ ]: [Text(0.5, 1.0, 'Unscaled')]
```



4 StandardScaler

[ ]:

```
from sklearn.preprocessing import StandardScaler
standardScaler = StandardScaler()
standardScaled = standardScaler.fit_transform(df)
df_StandardScaled = pd.DataFrame(standardScaled, columns = columnLabels)
df_StandardScaled
```

[ ]:

```
          Age  EstimatedSalary
0   -1.781797        -1.490046
1   -0.253587        -1.460681
2   -1.113206        -0.785290
3   -1.017692        -0.374182
4   -1.781797         0.183751
..        …              …
395  0.797057        -0.844019
396  1.274623        -1.372587
397  1.179110        -1.460681
398 -0.158074        -1.078938
399  1.083596        -0.990844

[400 rows x 2 columns]
```

[ ]:

```
print("Standard Scaled mean Age: ", np.rint(df_StandardScaled['Age'].mean()))
print("Standard Scaled mean EstimatedSalary: ", np.
 ↪rint(df_StandardScaled['EstimatedSalary'].mean()))
print("Standard Scaled variance of Age: ", np.rint(df_StandardScaled['Age'].
 ↪var()))
print("Standard Scaled variance of EstimatedSalary: ", np.
 ↪rint(df_StandardScaled['EstimatedSalary'].var()))
```

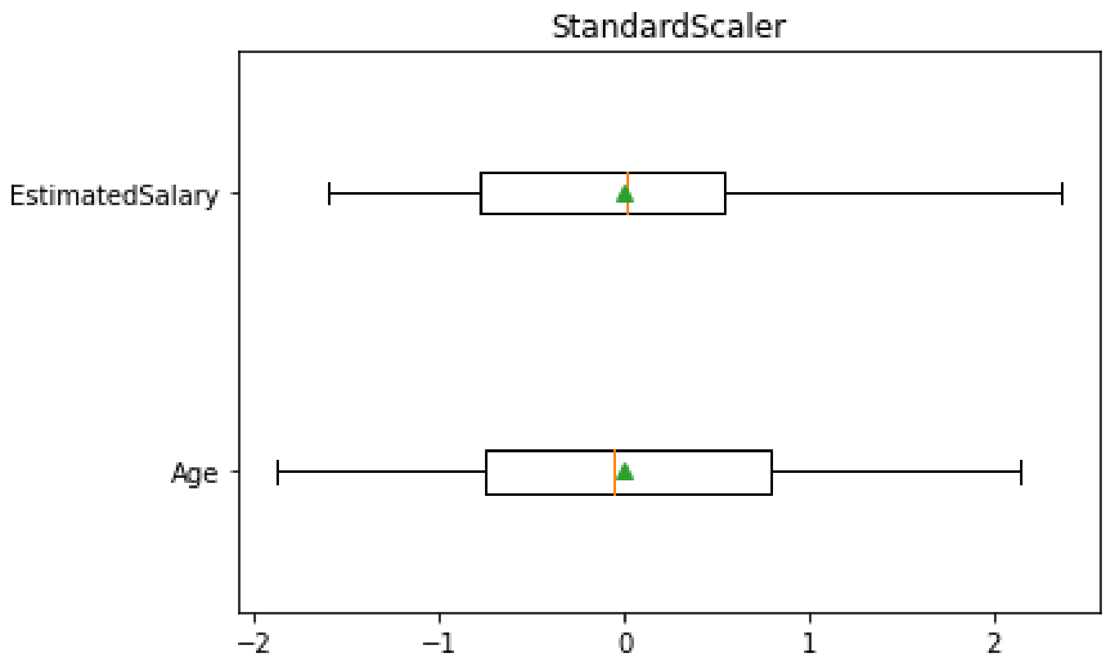
```
Standard Scaled mean Age:  -0.0
Standard Scaled mean EstimatedSalary:  -0.0
Standard Scaled variance of Age:  1.0
Standard Scaled variance of EstimatedSalary:  1.0
```

[ ]:

```
fig_StandardScaler, ax = plt.subplots(1, figsize = (6, 4))
ax.boxplot(df_StandardScaled, vert = False, showmeans = True, labels =␣
 ↪columnLabels)
ax.set(title = 'StandardScaler')
```

[ ]:

```
[Text(0.5, 1.0, 'StandardScaler')]
```



5 MinMaxScaler

[ ]:

```
from sklearn.preprocessing import MinMaxScaler
minMaxScaler = MinMaxScaler()
minMaxScaled = minMaxScaler.fit_transform(df)
df_MinMaxScaled = pd.DataFrame(minMaxScaled, columns = columnLabels)
df_MinMaxScaled
```

[ ]:

```
          Age  EstimatedSalary
0    0.023810         0.029630
1    0.404762         0.037037
2    0.190476         0.207407
3    0.214286         0.311111
4    0.023810         0.451852
..        …                …
395  0.666667         0.192593
396  0.785714         0.059259
397  0.761905         0.037037
398  0.428571         0.133333
399  0.738095         0.155556

[400 rows x 2 columns]
```

[ ]:

```
print('Minimum Age: ', np.rint(df_MinMaxScaled['Age'].min()))
print('Maximum Age: ', np.rint(df_MinMaxScaled['Age'].max()))
print('Minimum EstimatedSalary: ', np.rint(df_MinMaxScaled['EstimatedSalary'].
 ↪min()))
print('Maximum EstimatedSalary: ', np.rint(df_MinMaxScaled['EstimatedSalary'].
 ↪max()))
```

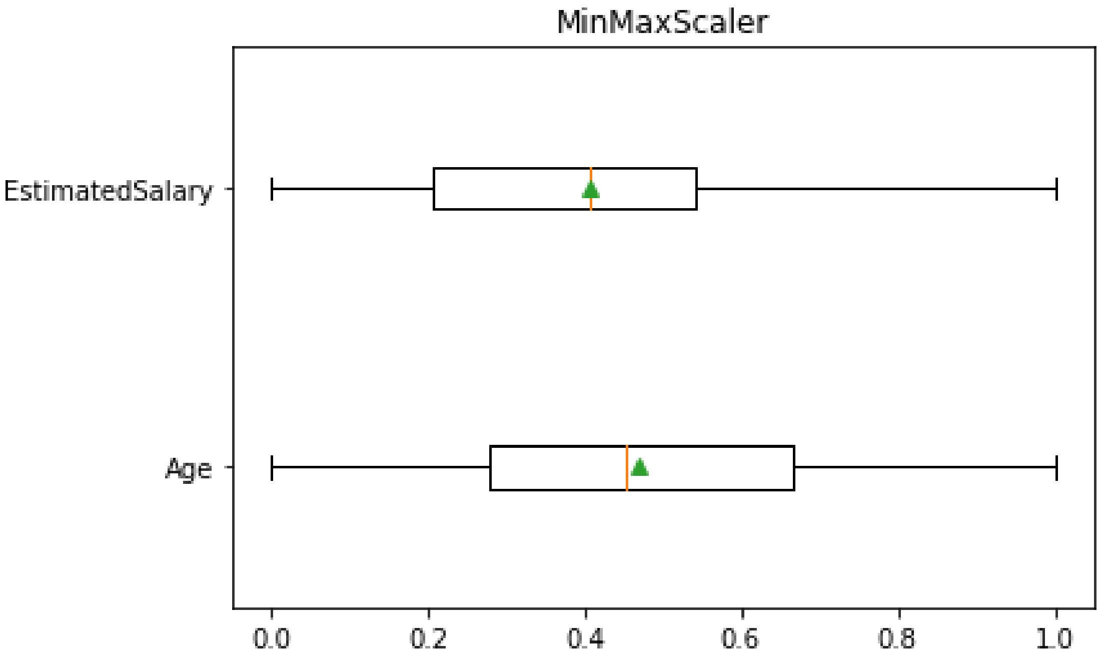
```
Minimum Age:  0.0
Maximum Age:  1.0
Minimum EstimatedSalary:  0.0
Maximum EstimatedSalary:  1.0
```

[ ]:

```
fig_MinMaxScaler, ax = plt.subplots(1, figsize = (6, 4))
ax.boxplot(df_MinMaxScaled, vert = False, showmeans = True, labels =␣
 ↪columnLabels)
ax.set(title = 'MinMaxScaler')
```

[ ]:

```
[Text(0.5, 1.0, 'MinMaxScaler')]
```



6 RobustScaler

[ ]:

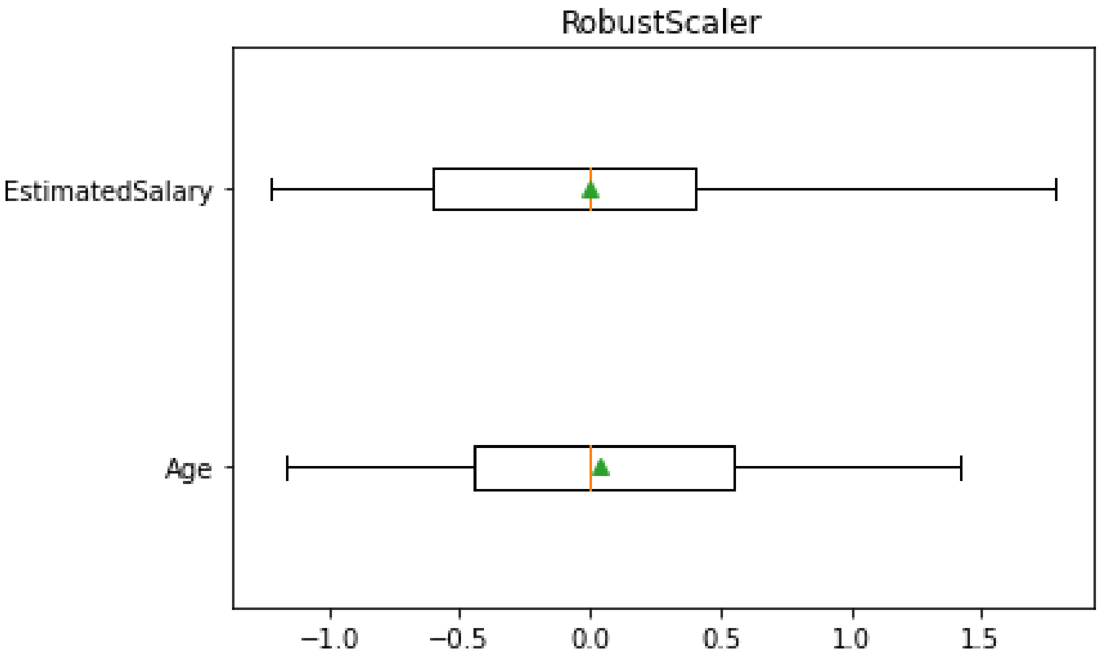
```
from sklearn.preprocessing import RobustScaler
robustScaler = RobustScaler()
robustScaled = robustScaler.fit_transform(df)
df_RobustScaled = pd.DataFrame(robustScaled, columns = columnLabels)
```

```
df_RobustScaled
```

```
[ ]:          Age  EstimatedSalary
     0   -1.107692        -1.133333
     1   -0.123077        -1.111111
     2   -0.676923        -0.600000
     3   -0.615385        -0.288889
     4   -1.107692         0.133333
     ..        …               …
     395  0.553846        -0.644444
     396  0.861538        -1.044444
     397  0.800000        -1.111111
     398 -0.061538        -0.822222
     399  0.738462        -0.755556

     [400 rows x 2 columns]
```

```
[ ]: fig_RobustScaler, ax = plt.subplots(1, figsize = (6, 4))
     ax.boxplot(df_RobustScaled, vert = False, showmeans = True, labels =␣
      ↪columnLabels)
     ax.set(title = 'RobustScaler')
```

```
[ ]: [Text(0.5, 1.0, 'RobustScaler')]
```



```
[ ]:
```