1 A4:Feature_Scaling : Demonstrate the purpose of feature scaling and show that feature scaling does not affect the distribution of the data

In [1]:

```
import numpy as np
import pandas as pd
```

In [4]:

```
df = pd.read_csv('Social_Network_Ads - Social_Network_Ads.csv',usecols=['Age','Estimated
df.head()
```

Out[4]:

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0

In [5]:

```
from sklearn.model_selection import train_test_split
```

In [7]:

```
x_train,x_test,y_train,y_test = train_test_split(df.drop('Purchased',axis=1),df['Purchase
```

In [8]:

```
x_train.shape
```

Out[8]:

```
(280, 2)
```

In [9]:

```
x_test.shape
```

Out[9]:

```
(120, 2)
```

In [11]:

```
from sklearn.preprocessing import StandardScaler
```

In [12]:

```
scaler = StandardScaler()
```

In [13]:

```
scaler.fit(x_train)
```

Out[13]:

```
StandardScaler()
```

In [14]:

```
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.fit_transform(x_test)
```

In [17]:

```
scaler.mean_
```

Out[17]:

```
array([3.71666667e+01, 6.95916667e+04])
```

In [18]:

```
x_train
```

Out[18]:

	Age	EstimatedSalary
92	26	15000
223	60	102000
234	38	112000
232	40	107000
377	42	53000
...	...	...
323	48	30000
192	29	43000
117	36	52000
47	27	54000
172	26	118000

280 rows × 2 columns

In [19]:

```
x_train_scaled
```

Out[19]:

```
array([[-1.1631724 , -1.5849703 ],
       [ 2.17018137,  0.93098672],
       [ 0.0133054 ,  1.22017719],
       [ 0.20938504,  1.07558195],
       [ 0.40546467, -0.48604654],
       [-0.28081405, -0.31253226],
       [ 0.99370357, -0.8330751 ],
       [ 0.99370357,  1.8563962 ],
       [ 0.0133054 ,  1.24909623],
       [-0.86905295,  2.26126285],
       [-1.1631724 , -1.5849703 ],
       [ 2.17018137, -0.80415605],
       [-1.35925203, -1.46929411],
       [ 0.40546467,  2.2901819 ],
       [ 0.79762394,  0.75747245],
       [-0.96709276, -0.31253226],
       [ 0.11134522,  0.75747245],
       [-0.96709276,  0.55503912],
```

In [21]:

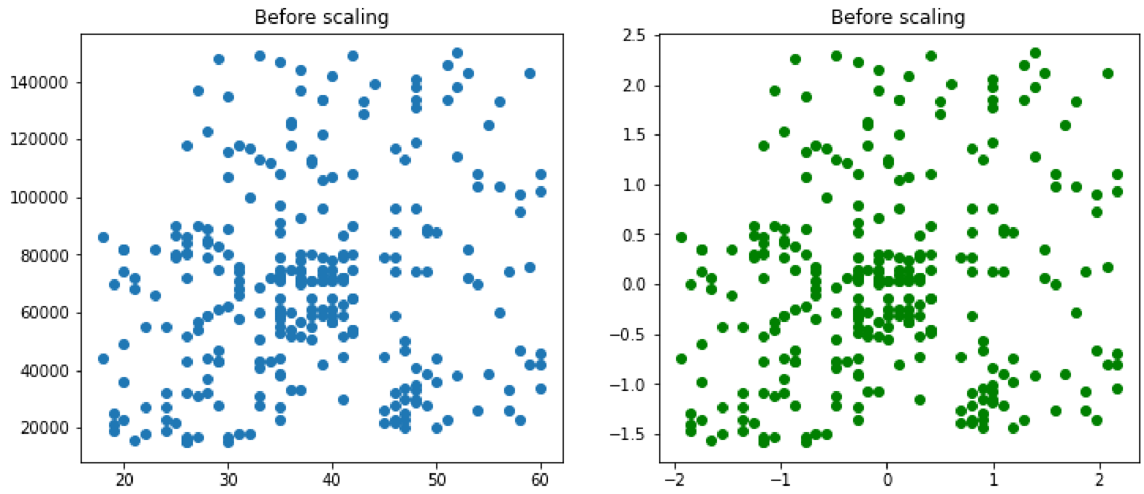
```
x_train_scaled = pd.DataFrame(x_train_scaled,columns=x_train.columns)
x_test_scaled = pd.DataFrame(x_test_scaled,columns=x_test.columns)
```

In [26]:

```
from matplotlib import pyplot as plt
fig,(ax1,ax2)= plt.subplots(ncols=2,figsize=(12,5))

ax1.scatter(x_train['Age'],x_train['EstimatedSalary'])
ax1.set_title('Before scaling')

ax2.scatter(x_train_scaled['Age'],x_train_scaled['EstimatedSalary'],color='green')
ax2.set_title('Before scaling')
plt.show()
```

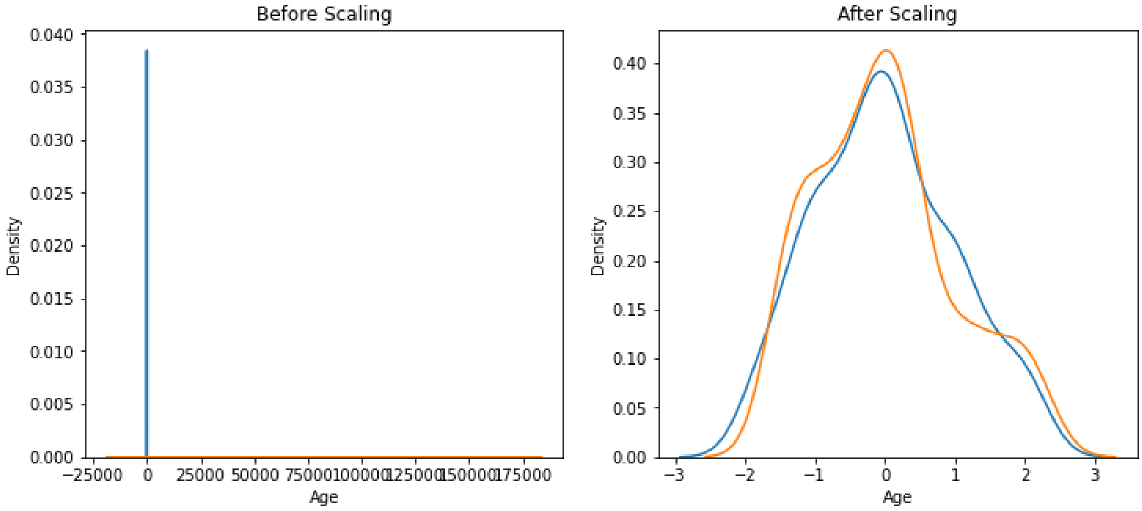


In [27]:

```
import seaborn as sns
fig,(ax1,ax2) = plt.subplots(ncols=2,figsize=(12,5))

ax1.set_title('Before Scaling')
sns.kdeplot(x_train['Age'],ax=ax1)
sns.kdeplot(x_train['EstimatedSalary'],ax=ax1)

ax2.set_title('After Scaling')
sns.kdeplot(x_train_scaled['Age'],ax=ax2)
sns.kdeplot(x_train_scaled['EstimatedSalary'],ax=ax2)
plt.show()
```



In [ ]:

```
```