**Code for** **both Standardization and Min-Max scaling:**

>>> x1 = [12, 14, 18, 23, 27, 28, 34, 37, 39, 40]

>>> x2 = [300, 500, 1000, 2000, 3500, 4000, 4300, 6000, 2500, 2700]

>>> x1

[12, 14, 18, 23, 27, 28, 34, 37, 39, 40]

>>> x2

[300, 500, 1000, 2000, 3500, 4000, 4300, 6000, 2500, 2700]

>>> import numpy as np

**Mean of x1:**

>>> np.mean(x1)

27.199999999999999

**mean of x2:**

>>> np.mean(x2)

2680.0

**Stanard deviation of x1**

>>> np.std(x1)

9.7652444925869624

**Standard deviation of x2:**

>>> np.std(x2)

1726.1517893858581

**Standardization of x1:**

>>> bhargavi = (x1-np.mean(x1))/np.std(x1)

>>> print(bhargavi)

[-1.55654065 -1.35173267 -0.94211671 -0.43009676 -0.0204808 0.08192319

0.69634713 1.0035591 1.20836708 1.31077107]

**Standardization of x2:**

>>> richa = (x2-np.mean(x2))/np.std(x2)

>>> print(richa)

[-1.37878952 -1.26292486 -0.97326319 -0.39393986 0.47504513 0.76470679

0.93850379 1.92335345 -0.1042782 0.01158647]

**Min-Max scaling of x1:**

>>> sai = (x1- np.min(x1))/(np.max(x1)-np.min(x1))

>>> print(sai)

[ 0. 0.07142857 0.21428571 0.39285714 0.53571429 0.57142857

0.78571429 0.89285714 0.96428571 1. ]

**Min-Max scaling of x2:**

>>> lokesh= (x2- np.min(x2))/(np.max(x2)-np.min(x2))

>>> print(lokesh)

[ 0. 0.03508772 0.12280702 0.29824561 0.56140351 0.64912281

0.70175439 1. 0.38596491 0.42105263]

**Screen shots of code and output for both Standardization and Min-Max scaling:**



