Experiment 2

Aim - Perform an experiment on binning equal width and binning equal depth.

Theory - Data binning, bucketing is a data pre-processing method used to minimize the effects of small observation errors. The original data values are divided into small intervals known as bins and then they are replaced by a general value calculated for that bin. This has a smoothing effect on the input data and may also reduce the chances of overfitting in the case of small datasets

There are 2 methods of dividing data into bins:

Equal Frequency Binning: bins have an equal frequency.

Equal Width Binning : bins have equal width with a range of each bin are defined as [min + w], [min + 2w] …. [min + nw] where w = (max – min) / (no of bins).

Program Code : (python)

# equal frequency

def equifreq(arr1, m):

a = len(arr1)

n = int(a / m)

for i in range(0, m):

arr = []

for j in range(i \* n, (i + 1) \* n):

if j >= a:

break

arr = arr + [arr1[j]]

print(arr)

# equal width

def equiwidth(arr1, m):

a = len(arr1)

w = int((max(arr1) - min(arr1)) / m)

min1 = min(arr1)

arr = []

for i in range(0, m + 1):

arr = arr + [min1 + w \* i]

arri=[]

for i in range(0, m):

temp = []

for j in arr1:

if j >= arr[i] and j <= arr[i+1]:

temp += [j]

arri += [temp]

print(arri)

# data to be binned

data = [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]

# no of bins

m = 3

print("equal frequency binning")

equifreq(data, m)

print("\n\nequal width binning")

equiwidth(data, 3)

Output:

