***Data Mining Assignment 3***

***Name : Venkatesh Suvarna***

***Student ID : 1001390986***

**Problem 3.3**

1. Smoothing by bin *means* – assume bin depth of 3 – each bin will contain 3 values

In bin means each value is replaced by the mean value of bin values.

1. Sorting in increasing order –

*13, 15, 16*, 16, 19, 20, *20, 21, 22*, 22, 25, 25, *25, 25, 30*, 33, 33, 35*, 35, 35, 35*, 36, 40, 45, *46, 52, 70*

1. Partitioning into bins of width = 3

Bin 1 : 13,15,16

Bin 2 : 16,19,20

Bin 3 : 20,21,22

Bin 4 : 22,25,25

Bin 5 : 25,25,30

Bin 6 : 33,33,35

Bin 7 : 35,35,35

Bin 8 : 36,40,45

Bin 9 : 46,52,70

1. Calculating mean values of each bin

Bin 1 = (13 + 15 + 16)/3 = 14.667 = **15** (rounding off)

Bin 2 = (16 + 19 + 20)/3 = 18.333 = **18**

Bin 3 = (20 + 21 + 22)/3 = **21**

Bin 4 = (22 + 25 + 25)/3 = **24**

Bin 5 = (25 + 25 + 30)/3 = 26.667 = **27**

Bin 6 = (33 + 33 + 35)/3 = 33.667 = **37**

Bin 7 = (35 + 35 + 35)/3 = **35**

Bin 8 = (36 + 40 + 45)/3 = 40.333 = **40**

Bin 9 = (46 + 52 + 70)/3 = **56**

1. Replacing the value of each bin’s mean to the respective bin

Bin 1 : 15,15,15

Bin 2 : 18,18,18

Bin 3 : 21,21,21

Bin 4 : 24,24,24

Bin 5 : 27,27,27

Bin 6 : 37,37,37

Bin 7 : 35,35,35

Bin 8 : 40,40,40

Bin 9 : 56,56,56

1. We can determine outliers by clustering the input data

For example – When similar values are grouped together, the values outside the set of groups can be considered as outliers.

1. Other methods for data smoothing are as follows
2. Regression – Linear or Multiple Linear
3. Binning Methods – Smoothing by bin median or Smoothing by bin boundaries
4. Outlier Analysis – values that belong to none of the clusters
5. Clustering – grouping the data to maximize the intraclass similarity and minimize the interclass similarity

**Problem 3.5**

1. Min-Max Normalization

It maps the value, Vi, of A to Vi’, in the range [new\_minA, new\_maxA]

1. Z-Score (Zero Mean) Normalization

We normalize the input data based on mean and standard deviation of A.

Range can be calculated as follows –

*[(min\_value – mean)/std\_deviation, (max\_value – mean)/std\_deviation]*

This practically can be anything between [ – ∞, ∞]

1. Z-Score using mean absolute deviation

This is just a variation of Z-Score Normalization, and uses absolute mean deviation to normalize the data.

Range can be calculated as follows –

[(min\_value – mean)/sA, (max\_value – mean)/sA]

This ranges from [ – ∞, ∞], just as the same for Z-Score Normalization.

1. Decimal Scaling

Normalizes by moving decimal point of values of attribute X. The formula for decimal scaling is Vj’ = Vj / 10j where j is the minimum integer for which max(|Vi’|)<1

The final range of values is from [-1, 1]

**Problem 3.7**

(a) Use min-max normalization to transform the value 35 for age onto the range [0.0,1.0]. For readability, let A be the attribute age. We have minA = 13, maxA = 70, new minA = 0,new maxA = 1.0, then v = 35 is transformed to **v0 = 0.39**.

(b) Use z-score normalization to transform the value 35 for age, where the standard deviation of age is 12.94 years. We have A = 809/27 = 29.96 and σA = 12.94, then v = 35 is transformed to **v0 = 0.39**.

(c) Use normalization by decimal scaling to transform the value 35 for age. We have j = 2, v = 35 is transformed to **v0 = 0.35**.

(d) Comment on which method you would prefer to use for the given data, giving reasons as to why.

Decimal scaling of normalization would be easier to calculate and interpret and would also maintain the data distribution. Min-max normalization is not adaptive to new values in the column that fall outside the current (min,max) of the column. We could use Z-score normalization too, but decimal scaling is easier to interpret.

**Problem 3.9**

**(a) equal-frequency partitioning**

Bin 1 : 5,10,11,13

Bin 2 : 15,35,50,55

Bin 3 : 72,92,204,215

**(b) equal-width partitioning**

The width of each interval is (215−5)/3 = 70. Therefore

Bin 1 : 5,10,11,13,15,35,50,55,72

Bin 2 : 92

Bin 3 : 204,215

**(c) clustering**

We will use a simple clustering technique:

We partition the data along the 2 biggest gaps in the data.

Bin 1 : 5,10,11,13,15

Bin 2 : 35,50,55,72,92

Bin 3 : 204,215

**Program – Binning and Normalization**

**Input Database – Wine Quality – Red**

**Output of Program**

1. **Normalization**

C:\Python35\python.exe "C:/Users/Venkatesh Suvarna/PycharmProjects/DataMining\_Assignment3/binning\_and\_normalization.py"

Press 1 to normalize Quality column.

Press 2 to bin the Quality column.1

0 0.4

1 0.4

2 0.4

3 0.6

4 0.4

5 0.4

6 0.4

7 0.8

8 0.8

9 0.4

10 0.4

11 0.4

12 0.4

13 0.4

14 0.4

15 0.4

16 0.8

17 0.4

18 0.2

19 0.6

20 0.6

21 0.4

22 0.4

23 0.4

24 0.6

25 0.4

26 0.4

27 0.4

28 0.4

29 0.6

...

1569 0.6

1570 0.6

1571 0.6

1572 0.4

1573 0.6

1574 0.6

1575 0.6

1576 0.6

1577 0.6

1578 0.6

1579 0.4

1580 0.6

1581 0.4

1582 0.4

1583 0.4

1584 0.8

1585 0.6

1586 0.6

1587 0.6

1588 0.6

1589 0.4

1590 0.6

1591 0.6

1592 0.6

1593 0.6

1594 0.4

1595 0.6

1596 0.6

1597 0.4

1598 0.6

Name: quality, dtype: float64

None

Process finished with exit code 0

**Output Note:** Here I have used min-max normalization to bring the range of the “Quality” column in my dataset within the range [0,1], since all the integer values in the original dataset were positive values.

1. **Binning**

C:\Python35\python.exe "C:/Users/Venkatesh Suvarna/PycharmProjects/DataMining\_Assignment3/binning\_and\_normalization.py"

Press 1 to normalize Quality column.

Press 2 to bin the Quality column.2

low 63

medium 681

high 638

very high 217

Name: quality, dtype: int64

Process finished with exit code 0

**Output Note:** Here the “Quality” column in the original database contains positive values, so I decided to divide the output values into 4 bin – denoting quality as Low, Medium, High and Very High and outputted the count of values in each of the bins.