Assignment 2 - (Spring 2021)
Due Date: Mar. 14, Sunday 11:59PM

1) In real-world data, tuples with *missing values* for some attributes are a common occurrence. Describe various methods for handling this problem.

Answer: There are six methods in which we can handle tuple missing values for attributes:

- Ignore the tuple: ignore that tuple that is a missing value. This method can be used when the class label is missing. “It is not effective unless the tuple contains several attributes with missing values.”[1]
- “Fill in the missing values manually”[1]: This approach is not feasible since we are dealing with a vast dataset, and if there are many missing values, it is very time-consuming to fill each missing value.
- “Use a global constant to fill the missing value”[1]: In this case, all the missing values are replaced with common constant values, such as label them as “unknown” or ∞.
- “Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value”[1]: Central tendency indicate the middle value, which can be used to deal with missing tuples value. If we have symmetric data distribution, we can use the mean value to fill the missing values. If the data distribution is skewed, the median can be used to serve the missing values.
- “Use the attribute mean or median for all samples belonging to the same class as the given tuple”[1]: if we are classifying any customers according to the credit risk, we may replace the missing value in the same credit risk category by its mean income. In case the data distribution is skewed, the median value is a better choice. [1]
- “Use the most probable value to fill in the missing value: the most probable value can be determined with regression, inference-based tools using a Bayesian formalism, or decision tree induction. Example: use other customers attribute and construct a decision tree to predict the missing value.”[1]

2. Suppose that the data for analysis includes the attribute *age*. The *age* values for the data tuples are (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.

(a) Use *smoothing by bin means* to smooth the above data, using a bin depth of 3. Illustrate your steps. Comment on the effect of this technique for the given data.

Answer:

With the smoothing by bin means method, we replace each value in the bin with the bin's mean value.

Bin depth = 3

Bin Number	Values in Bin	Mean of Bin	New value for Bin
1	13, 15, 16	$\frac{13+15+16}{3} = 14.67$	14.67,14.67,14.67
2	16, 19, 20	$\frac{16+19+20}{3} = 18.33$	18.33,18.33,18.33
3	20, 21, 22	$\frac{20+21+22}{3} = 21$	21,21,21
4	22, 25, 25	$\frac{22+25+25}{3} = 24$	24,24,24
5	25, 25, 30	$\frac{25+25+30}{3} = 26.67$	26.67,26.67,26.67
6	33, 33, 35	$\frac{33+33+35}{3} = 33.67$	33.67,33.67,33.67
7	35, 35, 35	$\frac{35+35+35}{3} = 35$	35,35,35
8	36, 40, 45	$\frac{36+40+45}{3} = 40.33$	40.33,40.33,40.33
9	46, 52, 70	$\frac{46+52+70}{3} = 56$	56,56,56

Here the value of data is the mean of 9 bins. In each bin, they are consulting their neighborhood and replacing the whole bin with its mean value. So overall, data has the same value in an interval of three. With this approach, there wouldn’t be any outliers. Bin Smoothing helps in removing noise from data and is known as local smoothing

(b) How might you determine *outliers* in the data?

Answer Identifying the outlier is an approach to identify the values that fall outside the group. The similar value in data is grouped together to form clusters. The value which is not falling in the group or outside the group is known as the value of the outlier.

An outlier can be identified in many ways. Below are some ways:

- Histograms: First, divide the data into equi-width histograms and identify the outlying value. The identified outlying value is outliers.
- By clustering: In this approach, the data having similar values are group together to form a cluster, and the values which do not fall in any cluster are outliers
- Fit the model: First, fit the model to data and at any point the threshold value with identifying the outlier.
-

(d) What other binning methods are there for *data smoothing*? Briefly describe how they work.

Answer: There are two more binning method for smoothing:

- Smoothing by bin medians: In this technique, each value within a bin is replaced by the median of that bin. So for part(a) of the question will give the following result with smoothing by bin median:

Bin Number	Values in Bin	New value for Bin replaced with median
1	13, 15, 16	15,15,15
2	16, 19, 20	19,19,19
3	20, 21, 22	21,21,21
4	22, 25, 25	25,25,25
5	25, 25, 30	25,25,25
6	33, 33, 35	33,33,33
7	35, 35, 35	35,35,35
8	36, 40, 45	40,40,40
9	46, 52, 70	52,52,52

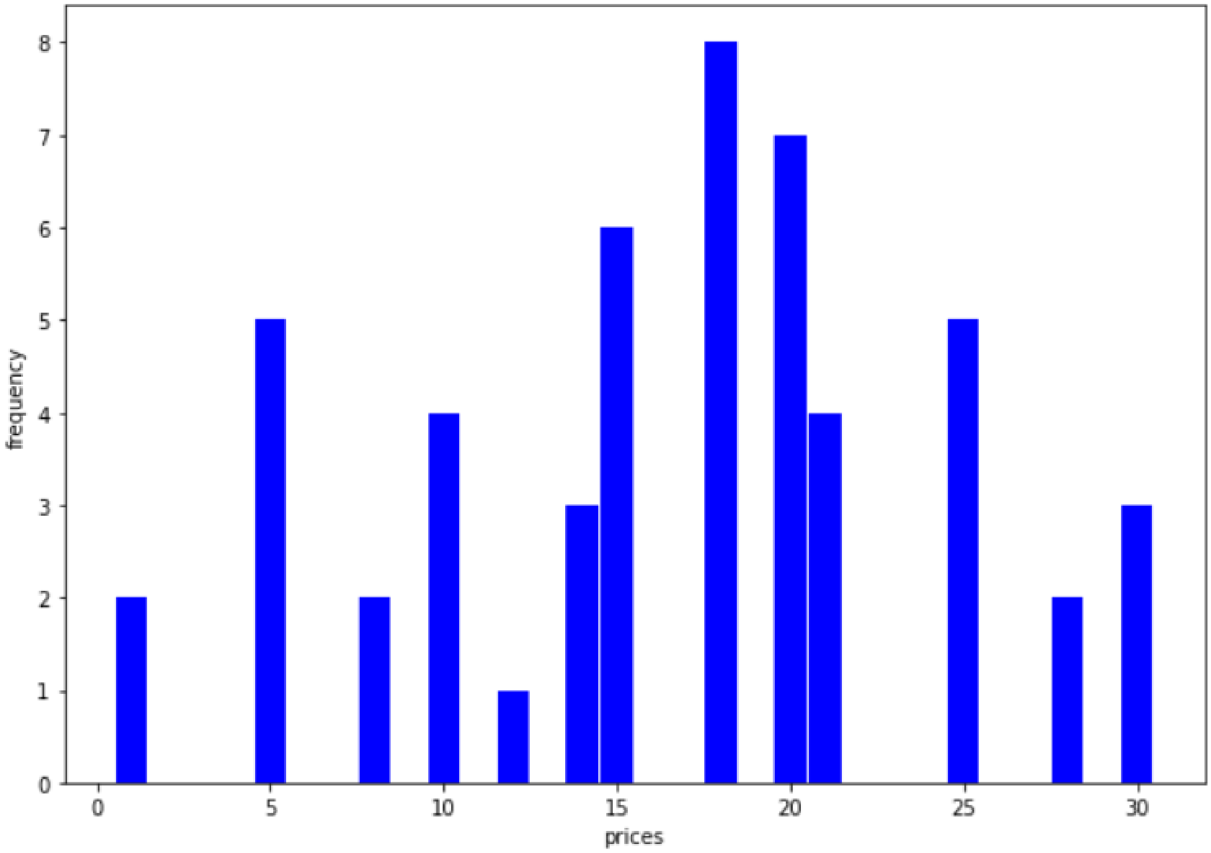
- Smoothing by bin boundaries: The bin value is replaced with the closet boundary(which is the minimum and maximum value of the bin)

Bin Numb	Values in Bin	New value for Bin replaced with median
1	13, 15, 16	13,16,15
2	16, 19, 20	16,20,20
3	20, 21, 22	20,20,22
4	22, 25, 25	22,25,25
5	25, 25, 30	25,25,30
6	33, 33, 35	33,33,35
7	35, 35, 35	35,35,35
8	36, 40, 45	36,36,45
9	46, 52, 70	46,46,70

3. The following are a list of prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.

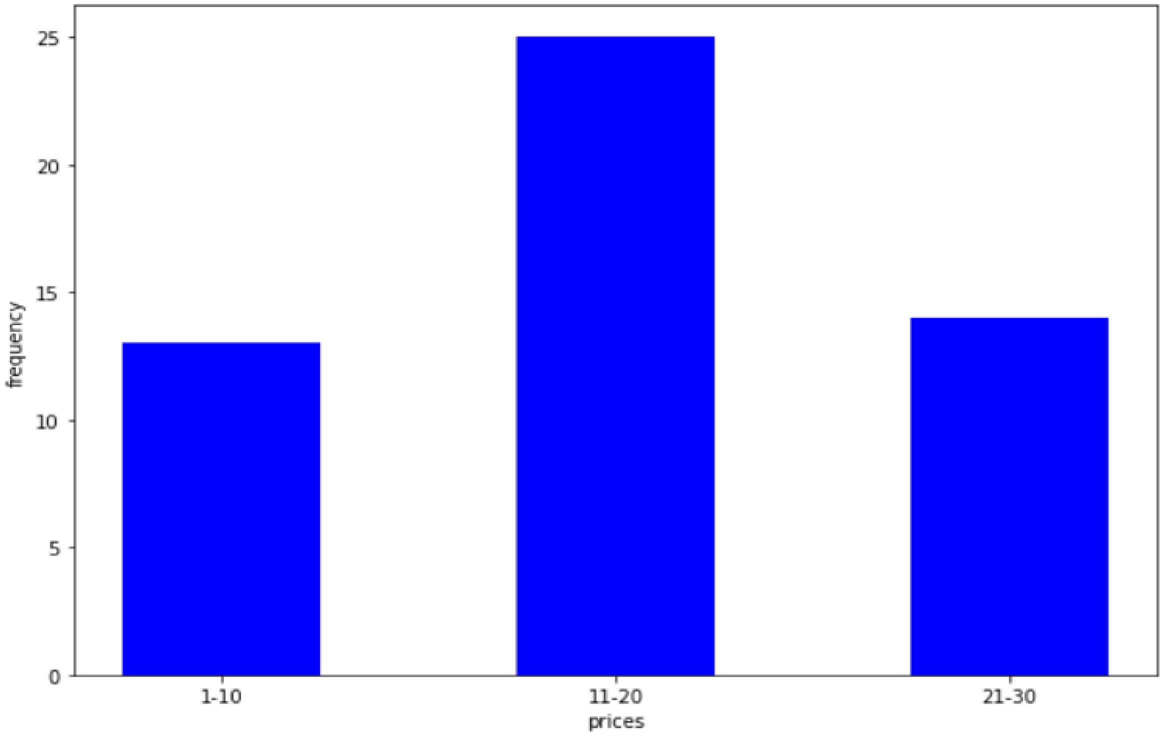
(a) Draw a histogram for the data using singleton buckets.

Answer:



(b) Draw a histogram for the data with each bucket denoting a continuous value range for the price attribute. For example, each bucket represents a different $10 range for *price.*

Answer:



4. Using the data for *age* and *body fat* given below, calculate the *correlation coefficient* (Pearson's product moment coefficient). Are these two variables positively or negatively correlated?

age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

Answer:

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B}$$

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\overline{AB}}{n\sigma_A\sigma_B}$$

n= number of tuples
$\bar{A}$ = mean of attribute A
$\bar{B}$ = mean of attribute B
σ_A = standard deviation for A attribute
σ_B = standard deviation for attribute B

age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

Now lets calculate :

$$\bar{A} = \frac{23+23+27+27+39+41+47+49+50+52+54+54+56+57+58+58+60+61}{18} = \frac{836}{18}$$

$$= 46.44$$

$$\bar{B} = \frac{\begin{gathered}9.5+26.5+7.8+17.8+31.4+25.9+27.4+27.2+31.2+34.6+42.5+28.8+33.4+30.2+34.1\\+32.9+41.2+35.7\end{gathered}}{18}$$

$$= \frac{518.1}{18} = 28.78$$

$$\sigma_A = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i^2 - \bar{A}^2)}$$

$\sigma_A =$

$$\sqrt{\frac{\begin{gathered}(23^2-44.66^2)+(23^2-44.66^2)+(27^2-44.66^2)+(27^2-44.66^2)+(39^2-44.66^2)+(41^2-44.66^2)+(47^2-44.66^2)+(49^2-44.66^2)+\\(50^2-44.66^2)+(52^2-44.66^2)+(54^2-44.66^2)+(54^2-44.66^2)+(56^2-44.66^2)+(57^2-44.66^2)+(58^2-44.66^2)+(58^2-44.66^2)+\\(60^2-44.66^2)+(61^2-44.66^2)\end{gathered}}{18}}$$

$$= 12.85$$

$$\sigma_B = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i^2 - \bar{B}^2)}$$

$\sigma_B =$

$$\sqrt{\frac{\begin{gathered}(9.5^2-28.78^2)+(26.5^2-28.78^2)+(7.8^2-28.78^2)+(17.8^2-28.78^2)+(31.4^2-28.78^2)+(25.9^2-28.78^2)+(27.4^2-28.78^2)+(27.2^2-28.78^2)\\+\\(31.2^2-28.78^2)+(34.6^2-28.78^2)+(42.5^2-28.78^2)+(28.8^2-28.78^2)+(33.4^2-28.78^2)+(30.2^2-28.78^2)+(34.1^2-28.78^2)+\\(32.9^2-28.78^2)+(41.2^2-28.78^2)+(35.7^2-28.78^2)\end{gathered}}{18}}$$

= 8.99

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\overline{AB}}{n\sigma_A \sigma_B}$$

$$r_{A,B} = \frac{(23\times 9.5 + 23\times 26.5 + 27\times 7.8 + 27\times 17.8 + 39\times 31.4 + 41\times 25.9 + 47\times 27.4 + 49\times 27.2 + 50\times 31.2 + 52\times 34.6 + 54\times 42.5 + 54\times 28.8 + 56\times 33.4 + 57\times 30.2 + 58\times 34.1 + 58\times 32.9 + 60\times 41.2 + 61\times 35.7) - 18(46.44\times 28.78)}{18(12.85\times 8.99)}$$

= 0.82

Since $r_{A,B} > 0$, A and B are positively correlated.

5. You are given a set of *m* objects that is divided into *K* groups, where the i^{th} group is of size m_i. If the goal is to obtain a sample of size $n < m$, what is the difference between the following two sampling schemes? (Assume sampling with replacement.)

(a) We randomly select $n * m_i / m$ elements from each group.

Answer:

In this, the elements are randomly distributed, having each group. So, each group member is equally distributed.

(b) We randomly select *n* elements from the data set, without regard for the group to which an object belongs.

Answer:

In this approach, to achieve equal distribution of each group, the n elements must be equal to m, which is the group size. Although this too doesn’t assure that the final distribution is having a sample from each group equally.

6. Discuss various ways of using sampling to reduce the number of data objects that need to be processed. Would simple random sampling (without replacement) be a good approach to sampling? Why or why not?

Answer:

Sampling is a technique where we select data from the population dataset for further study. This is a data reduction technique used for converting large dataset D to much smaller random dataset s. Below are a few techniques:

- Simple random sample without replacement (SRSWOR): Suppose we have a dataset D with N number of a tuple, to reduce the dataset to the size of *s*, simple random sample without replacement draw a tuple from D, where the probability of each tuple draw is 1/N.
- Simple random sample with replacement (SRSWR): this is similar to SRSWOR; the only difference is that every time a tuple is drawn from D, it is recorded and then replace within D. By doing so, it allows tuple to be drawn again. Here the final dataset s might have duplicate values.
- Cluster Sample: If the tuples are grouped into N different clusters, then a simple random sample of size s can be achieved, where sample size s is smaller than the cluster. “For example, tuples in a database have usually retrieved a page at a time, so that each page can be considered a cluster.”[1]
- Stratified sample: “If dataset D is divided parts which are mutually disjoint called strata, a stratified sample can be achieved by obtaining an SRS on each stratum. This helps to ensure a representative sample, especially when the data are skewed.”[1]

Yes, in my opinion, simple random sampling without replacement is a good approach. We are choosing the sample smaller dataset on a random basis which will give each tuple an equal opportunity to be picked for a sample. The final dataset will have a randomly selected dataset; unlike a simple random sample with replace, there are no chances of duplicate tuples. This will produce the subset which is balance since the selection is random. This is an un-bias sample dataset from D. Another advantage of this approach is that it is simple; it is randomly picking the tuples. However, this approach is not good if we have D(a large dataset that needs to convert) already small.

7. Use the methods below to *normalize* the following group of data: 200, 300, 400, 600, 1000

(a) min-max normalization by setting *min* = 0 and *max* = 1

Answer:

A = 200,300,400,600,1000

min_{new} = 0
max_{new} = 1
min_A= 200
max_A = 1000

$$v' = \frac{v-min_A}{max_A-min_A}(max_{new} - min_{new}) + min_{new}$$

for v= 200

$$v' = \frac{200-200}{1000-200}(1-0)+0 = 0$$

For v=300

$$v' = \frac{300-200}{1000-200}(1-0)+0 = 0.125$$

For v = 400

$$v' = \frac{400-200}{1000-200}(1-0)+0 = 0.25$$

For v = 600

$$v' = \frac{600-200}{1000-200}(1-0)+0 = 0.5$$

For v = 1000

$$v' = \frac{1000-200}{1000-200}(1-0)+0 = 1$$

Before Normalization A value	After Normalization A value
200	0
300	0.125
400	0.25
600	0.5
1000	1

(b) z-score normalization

Answer:

Below is the formula for calculating z-score normalization

v'=$\frac{v-\mu_A}{\sigma_A}$ where $\mu_A = Mean\ of\ A\ and\ \sigma_A\ is\ standard\ deviation\ of\ A$

$$\mu_A = \frac{200+300+400+600+1000}{5} = 500$$

$$\sigma_A = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i^2 - \bar{A}^2)}$$

σ_A=

$$\sqrt{\frac{1}{5}[(200^2-500^2)+(300^2-500^2)+(400^2-500^2)+(600^2-500^2)+(1000^2-500^2)]}$$

=282.84

Now lets calculate v’ for each v

For V = 200

v'=$\frac{200-500}{282.84} = \ -1.06$

For V = 300

v'=$\frac{300-500}{282.84} = -0.70$

For v = 400

v'=$\frac{400-500}{282.84} = -0.35$

For v= 600

v'=$\frac{600-500}{282.84} = 0.35$

For v =1000

v'=$\frac{1000-500}{282.84} = 1.77$

Before Normalization A value	After Normalization A value
200	-1.06
300	-0.70
400	-0.35
600	0.35
1000	1.77

(c) z-score normalization using the mean absolute deviation instead of standard deviation

z-score normalization using the mean absolute deviation is a variation of *z-score normalization* by replacing the standard deviation with the mean absolute deviation of *A*, denoted by S_A, which is

$$s_A = \frac{1}{n}(|v_1 - \bar{A}| + |v_2 - \bar{A}| + \ldots + |v_n - \bar{A}|)$$

Answer:

Below is the formula for calculating z-score normalization

v'=$\frac{v-\bar{A}}{s_A}$ where $\bar{A} = Mean\ of\ A\ and$

$s_A\ is\ absolute\ standard\ deviation\ of\ A$

$$\bar{A} = \frac{200+300+400+600+1000}{5} = 500$$

$$s_A = \frac{1}{n}(|v_1 - \bar{A}| + |v_2 - \bar{A}| + |v_3 - \bar{A}| + |v_4 - \bar{A}| + |v_{15} - \bar{A}|)$$

$$s_A = \frac{1}{5}(|200 - 500| + |300 - 500| + |400 - 500| + |600 - 500| + |1000 - 500|)$$

s_A= 240

Now lets calculate v’

For v = 200

$$v' = \frac{200-500}{240} = -1.25$$

For v = 300

$$v' = \frac{300-500}{240} = -0.83$$

For v = 400

$$v' = \frac{400-500}{240} = -0.42$$

For v = 600

$$v' = \frac{600-500}{240} = 0.42$$

For v=1000

$$v' = \frac{1000-500}{240} = 2.08$$

Before Normalization A value	After Normalization A value
200	-1.25
300	-0.83
400	-0.42
600	0.42
1000	2.08

(c) normalization by decimal scaling

Answer:

To calculate decimal scaling normalization we have below formula:

$$v'_i = \frac{v_i}{10^j} \quad where\ j\ is\ smallest\ integer\ such\ that\ \max(|v'_i|) < 1$$

So for v= 200

$$v' = \frac{200}{10^3} = 0.02 \qquad here\ j = 4$$

for v= 300

$$v' = \frac{300}{10^4} = 0.03 \qquad here\ j = 4$$

for v= 400

$$v' = \frac{400}{10^4} = 0.04 \qquad here\ j = 4$$

for v= 600

$$v' = \frac{600}{10^4} = 0.06 \qquad here\ j = 4$$

for v= 1000

$$v' = \frac{1000}{10^4} = 0.1 \qquad here\ j = 4$$

Before Normalization A value	After Normalization A value
200	0.02
300	0.03
400	0.04
600	0.06
1000	0.1

8) Consider a document-term matrix, where tf_{ij} is the frequency of the i^{th} word(term) in the j^{th} document and m is the number of documents. Consider the variable transformation that is defined by

$$tf'_{ij} = tf_{ij} * \log \frac{m}{df_i}$$

where df_i is the number of documents in which the i^{th} term appears and is known as the **document frequency** of the term. This transformation is known as the **inverse document frequency** transformation.

(a) What is the effect of this transformation if a term occurs in one document? In every document?

Answer:

In this situation, the term's effect if it appears only in one document is high, which means that the term will be weighted highly.

let's show it with an example:

Let the frequency be tf_{ij}: =2

The total document to be m =6

The number of the document in which term appears is df_i = 1

With the given formula, the inverse document frequency transformation would be:

$$2 \times \log\left(\frac{6}{1}\right) = 1.57$$

When the term appears in every document, the inverse transformation tends to 0
Lets show:
Let the frequency be tf_{ij}: =2
The total document to be m =6
The number of the document in which term appears is df_i = 6
With the given formula, the inverse document frequency transformation would be:

$$2 \times \log\left(\frac{6}{6}\right) = 0$$

Therefore the transformation of the term if it appears in one document is high.

(b) What might be the purpose of this transformation?
Answer:

Since the data has grown vastly and searched for a specific word in this extensive database for a common term that may appear in several documents is the primary purpose of this transformation. With this approach, a user may find it easier to locate their interest. Suppose from college datasets we want students for a basketball team. We can search through student profiles having the word "basketball" in the documents of a student. The word basketball is certainly interest for this purpose, but the word "the" is not.