

Lesson 3

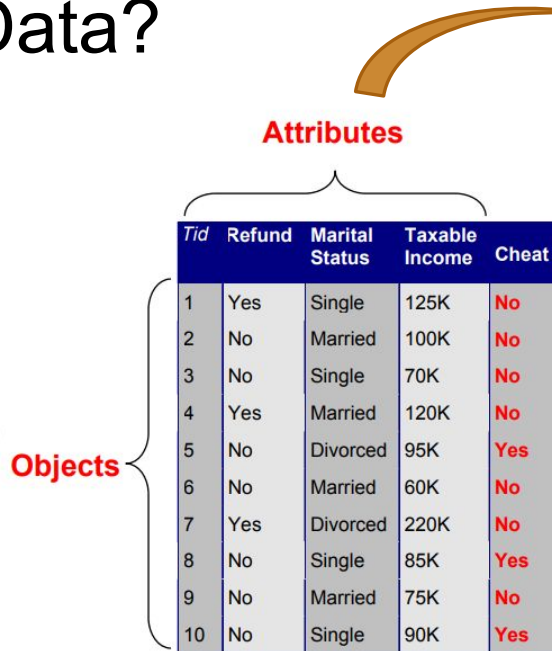
Preprocessing

What is Data?

- Collection of data objects and their attributes

- An attribute is a property or characteristic of an object
 - Examples: eye color of a person, temperature, etc.
 - Attribute is also known as variable, field, characteristic, or feature

- A collection of attributes describe an object
 - Object is also known as record, point, case, sample, entity, or instance



Attributes

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Objects

- ▶ **Nominal (Categorical):** categories, states, or “names of things”

- ▶ Hair_color = {auburn, black, blond, brown, grey, red, white}
- ▶ marital status, occupation, ID numbers, zip codes
- ▶ Often attributes with “yes” and “no” as values
- ▶ Binary
 - ▶ Nominal attribute with only 2 states (0 and 1)

- ▶ **Ordinal**

- ▶ Values have a meaningful order (ranking) but magnitude between successive values is not known.
- ▶ Size = {small, medium, large}, grades, army rankings
- ▶ Month = {jan, feb, mar, ... }

- ▶ **Numeric**

- ▶ Quantity (integer or real-valued)
- ▶ Could also be intervals or ratios

What is Data?

- Collection of data objects and their attributes
- An attribute is a property or characteristic of an object
 - Examples: eye color of a person, temperature, etc.
 - Attribute is also known as variable, field, characteristic, or feature
- A collection of attributes describe an object
 - Object is also known as record, point, case, sample, entity, or instance

Attributes

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10				

Objects

- ▶ **Nominal (Categorical):** categories, states, or “names of things”
 - ▶ Hair_color = {auburn, black, blond, brown, grey, red, white}
 - ▶ marital status, occupation, ID numbers, zip codes
 - ▶ Often attributes with “yes” and “no” as values
 - ▶ Binary
 - ▶ Nominal attribute with only 2 states (0 and 1)
- ▶ **Ordinal**
 - ▶ Values have a meaningful order (ranking) but magnitude between successive values is not known.
 - ▶ Size = {small, medium, large}, grades, army rankings
 - ▶ Month = {jan, feb, mar, ... }
- ▶ **Numeric**
 - ▶ Quantity (integer or real-valued)
 - ▶ Could also be intervals or ratios

☒ **Nominal** — values from an unordered set

☒ **Ordinal** — values from an ordered set

☒ **Continuous** — real numbers

Why Data Preprocessing?

- Data in the real world is dirty
 - **incomplete**: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 - ◆ e.g., occupation=" "
 - **noisy**: containing errors or outliers
 - ◆ e.g., Salary="-10"
 - **inconsistent**: containing discrepancies in codes or names
 - ◆ e.g., Age="42" Birthday="03/07/1997"
 - ◆ e.g., Was rating "1,2,3", now rating "A, B, C"
 - ◆ e.g., discrepancy between duplicate records

Data Cleaning

- Importance
 - garbage in garbage out principle (GIGO)
- Data cleaning tasks
 - Fill in missing values
 - Identify outliers and smooth out noisy data
 - Correct inconsistent data
 - Resolve redundancy caused by data integration

How to handle missing data?

X1	X2	X3	X4	Y
78.5	67	1	0.2	73.2
78.5	67	0	0.2	69.2
78.5	67	0	0.2	69
78.5		0	0.2	69
75.5	66.5	1	0.2	73.5
75.5	66.5	1	0.4	
75.5	66.5	0	0.3	65.5
75.5	66.5	0	0.2	65.5
75		1		71
75	64	0	0.1	68
75	64	1	0.2	70.5

Option1: Remove rows with missing data

Not a good solution !

X1	X2	X3	X4	Y
78.5	67	1	0.2	73.2
78.5	67	0	0.2	69.2
78.5	67	0	0.2	69
78.5		0	0.2	69
75.5	66.5	1	0.2	73.5
75.5	66.5	1	0.4	
75.5	66.5	0	0.3	65.5
75.5	66.5	0	0.2	65.5
75		1		71
75	64	0	0.1	68
75	64	1	0.2	70.5

X1	X2	X3	X4	Y
78.5	67	1	0.2	73.2
78.5	67	0	0.2	69.2
78.5	67	0	0.2	69
75.5	66.5	1	0.2	73.5
75.5	66.5	0	0.3	65.5
75.5	66.5	0	0.2	65.5
75	64	0	0.1	68
75	64	1	0.2	70.5

Option 2: Fill by column average (mean)

X1	X2	X3	X4	Y
78.5	67	1	0.2	73.2
78.5	67	0	0.2	69.2
78.5	67	0	0.2	69
78.5		0	0.2	69
75.5	66.5	1	0.2	73.5
75.5	66.5	1	0.4	
75.5	66.5	0	0.3	65.5
75.5	66.5	0	0.2	65.5
75		1		71
75	64	0	0.1	68
75	64	1	0.2	70.5

X1	X2	X3	X4	Y
78.5	67	1	0.2	73.2
78.5	67	0	0.2	69.2
78.5	67	0	0.2	69
78.5	66.1	0	0.2	69
75.5	66.5	1	0.2	73.5
75.5	66.5	1	0.4	69.44
75.5	66.5	0	0.3	65.5
75.5	66.5	0	0.2	65.5
75	66.1	1	0.22	71
75	64	0	0.1	68
75	64	1	0.2	70.5

Option 3: Fill by linear interpolation

X1	X2	X3	X4	Y
78.5	67	1	0.2	73.2
78.5	67	0	0.2	69.2
78.5	67	0	0.2	69
78.5		0	0.2	69
75.5	66.5	1	0.2	73.5
75.5	66.5	1	0.4	
75.5	66.5	0	0.3	65.5
75.5	66.5	0	0.2	65.5
75		1		71
75	64	0	0.1	68
75	64	1	0.2	70.5

X1	X2	X3	X4	Y
78.5	67	1	0.2	73.2
78.5	67	0	0.2	69.2
78.5	67	0	0.2	69
78.5	66.75	0	0.2	69
75.5	66.5	1	0.2	73.5
75.5	66.5	1	0.4	69.5
75.5	66.5	0	0.3	65.5
75.5	66.5	0	0.2	65.5
75	65.25	1	0.15	71
75	64	0	0.1	68
75	64	1	0.2	70.5

Option 4: Fill by global constant (i.e. '0' or NA)

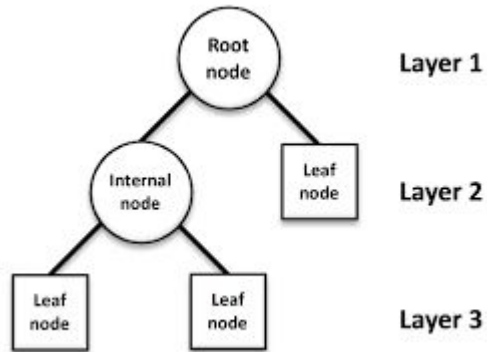
X1	X2	X3	X4	Y
78.5	67	1	0.2	73.2
78.5	67	0	0.2	69.2
78.5	67	0	0.2	69
78.5		0	0.2	69
75.5	66.5	1	0.2	73.5
75.5	66.5	1	0.4	
75.5	66.5	0	0.3	65.5
75.5	66.5	0	0.2	65.5
75		1		71
75	64	0	0.1	68
75	64	1	0.2	70.5

X1	X2	X3	X4	Y
78.5	67	1	0.2	73.2
78.5	67	0	0.2	69.2
78.5	67	0	0.2	69
78.5	0	0	0.2	69
75.5	66.5	1	0.2	73.5
75.5	66.5	1	0.4	0
75.5	66.5	0	0.3	65.5
75.5	66.5	0	0.2	65.5
75	0	1	0	71
75	64	0	0.1	68
75	64	1	0.2	70.5

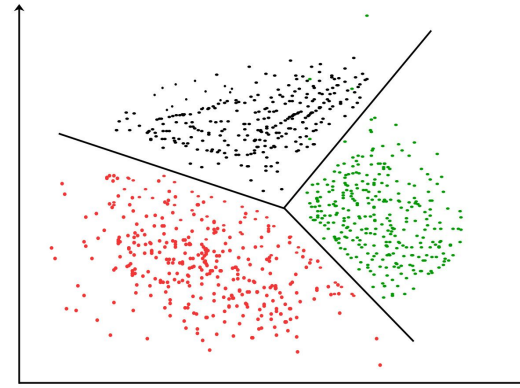
Option 5: by most **probable** value

Prediction by means of

Decision Tree



Clustering



Data Transformation by Binning

Some data mining algorithms require that the data be in the form of a categorical attribute, thus it is often necessary to transform a continuous attribute into a categorical attribute

- **Binning** or discretization

- is the process of transforming numerical variables into categorical counterparts.
 - Example: to **bin** values for Age into categories such as 20-39, 40-59, and 60-79
- There are two types of binning
 - Unsupervised
 - Supervised

Unsupervised Binning

- Unsupervised binning methods transform numerical variables into categorical counterparts but do not use the target (class) information. *Equal Width* and *Equal Frequency* are two unsupervised binning methods.

1- Equal Width Binning

The algorithm divides the data into k intervals of equal size. The width of intervals is:

$$w = (max-min)/k$$

And the interval boundaries are:

$$min+w, min+2w, \dots, min+(k-1)w$$

2- Equal Frequency Binning

The algorithm divides the data into k groups which each group contains approximately same number of values. For the both methods, the best way of determining k is by looking at the histogram and try different intervals or groups.

Example

- **Data :** 0, 4, 12, 16, 16, 18, 24, 26, 28

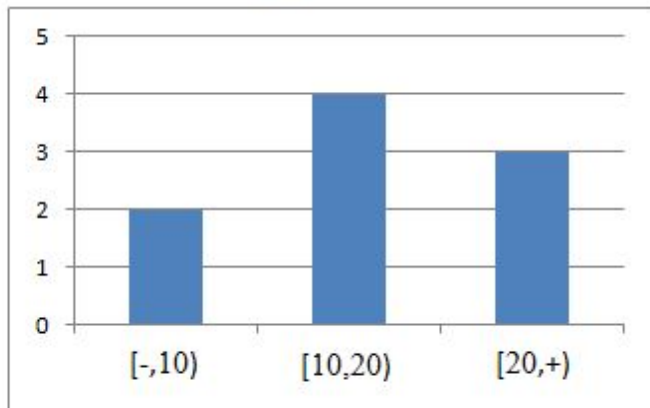
- **Equal width**

- Bin 1: 0, 4 $[-,10)$
- Bin 2: 12, 16, 16, 18 $[10,20)$
- Bin 3: 24, 26, 28 $[20,+)$

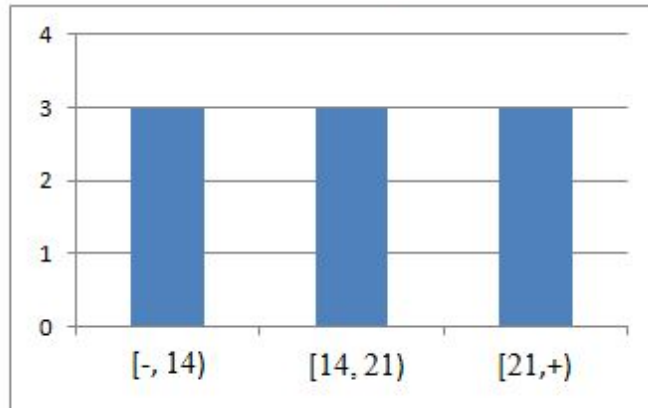
- **Equal frequency**

- Bin 1: 0, 4, 12 $[-, 14)$
- Bin 2: 16, 16, 18 $[14, 21)$
- Bin 3: 24, 26, 28 $[21,+)$

Equal width



Equal frequency



Data transformation by smoothing (noisy data handling)

Sorted data for price (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

- Partition into 3 (equal-frequency) bins:

- Bin 1: 4, 8, 15
- Bin 2: 21, 21, 24
- Bin 3: 25, 28, 34

- Smoothing by bin means:

- Bin 1: 9, 9, 9
- Bin 2: 22, 22, 22
- Bin 3: 29, 29, 29

- Smoothing by bin boundaries:

- Bin 1: 4, 4, 15
- Bin 2: 21, 21, 24
- Bin 3: 25, 25, 34

- Smoothing by bin median

- Bin 1: 8, 8, 8
- Bin 2: 21, 21, 21
- Bin 3: 28, 28, 28

Supervised Binning

- Supervised binning methods transform numerical variables into categorical counterparts and refer to the target (class) information when selecting discretization cut points.
- *Entropy-based* binning is an example of a supervised binning method.

Entropy-based Binning

Entropy based method uses a split approach. The entropy (or the information content) is calculated based on the class label. Intuitively, it finds the best split so that the bins are as pure as possible that is the majority of the values in a bin correspond to have the same class label. Formally, it is characterized by finding the split with the maximal information gain.

Seatwork

- Discretize the following values using **Equal-width** and **Equal-depth** binning using 3 bins

13 15 16 16 17 19 20 21 22 22 25 30 33 35 35 36 40 45 47

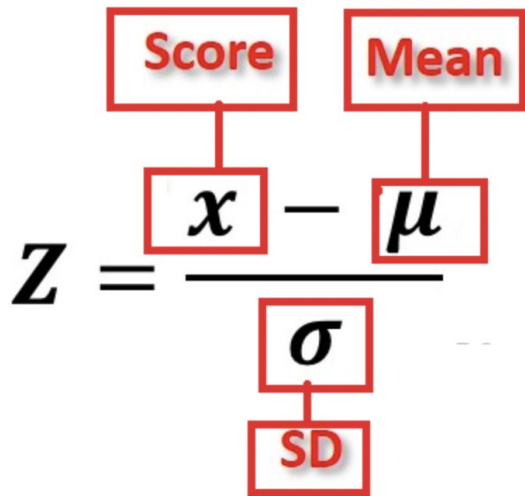
- Smooth each bin by bin means, bin boundaries, bin median

Data Transformation on numerical data / Data Normalization

One of the common challenges is that, usually, databases contain attributes of different units, range, and scales. Applying algorithms to such drastically ranging data may not deliver accurate results. This calls for **data normalization** in data mining.

It is a necessary process required to normalize heterogeneous data. Data can be put into a smaller range, such as 0.0 to 1.0 or -1.0 to 1.0. In simple words, data normalization makes data easier to classify and understand.

Method 1: Z-score normalization



The diagram illustrates the Z-score formula with labels in red boxes connected to the variables by lines. The formula is $Z = \frac{x - \mu}{\sigma}$. The label 'Score' points to x , 'Mean' points to μ , and 'SD' (Standard Deviation) points to σ .

$$Z = \frac{x - \mu}{\sigma}$$

The z-score is very useful when we are understanding the data. The z-score is a very useful statistic of the data due to the following facts;

- It allows a data administrator to understand the probability of a score occurring within the normal distribution of the data.
 - a **high z-score** value means a very **low probability** of data **above** this z-score.
 - a **low z-score** value means a very **low probability** of data **below** this z-score.
- The z-score enables a data administrator to compare two different scores that are from different normal distributions of the data.

What does the positive and negative z-score mean?

- The **standard deviation of the z-scores is always 1** and similarly, the **mean of the z-scores is always 1**.
- Z-scores values **above the 0** represent that sample values **are above the mean**.
- z-scores values **below the 0** represent that sample values **are below the mean**.

How to calculate Z-Score of the following data?

marks	marks after z-score normalization
8	-1.14
10	-0.7
15	0.3
20	1.4

$$ZScore = \frac{x - \mu}{\sigma} = \frac{8 - 13.25}{4.6} = -1.14$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{10 - 13.25}{4.6} = -0.7$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{15 - 13.25}{4.6} = 0.3$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{20 - 13.25}{4.6} = 1.4$$

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

$$\text{Standard deviation} = \sqrt{\frac{\sum (\text{every individual value of marks} - \text{mean of marks})^2}{n}}$$

$$\text{Mean of marks} = 8 + 10 + 15 + 20 / 4 = 13.25$$

$$= \sqrt{\frac{(8 - 13.25)^2 + (10 - 13.25)^2 + (15 - 13.25)^2 + (20 - 13.25)^2}{4}}$$

$$= \sqrt{\frac{(-5.25)^2 + (-3.25)^2 + (1.75)^2 + (6.75)^2}{4}}$$

$$= \sqrt{\frac{27.56 + 10.56 + 3.06 + 45.56}{4}} = \sqrt{\frac{86.74}{4}} = \sqrt{21.6} = 4.6$$

Z-score

- Z-score normalization (μ : mean, σ : standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

- Ex. Let $\mu = 54,000$, $\sigma = 16,000$. Then

$$\frac{73,600 - 54,000}{16,000} = 1.225$$

	X1	X2	X3	X4		X1	X2	X3	X4	
	-0.26	101.87	-9.2	19593.5		-0.74	1.44	0.16	2.12	
	1.26	102.23	-71.7	3899.2		1.07	1.70	-0.42	-0.07	
	-0.47	99.78	113.9	-4608.1		-0.99	-0.09	1.29	-1.26	
	1.61	101.07	41	888		1.49	0.85	0.62	-0.49	
	-0.41	98.83	156.1	4810.4		-0.92	-0.78	1.68	0.05	
	1.63	99.92	-3.5	5490.1		0.61	-0.89	-0.77	-1.61	
	0.87	98.67	-109.7	-7091.2		0.61	-0.89	-0.77	-1.61	
	0.05	100.51	-23.6	9864.5		-0.37	0.45	0.03	0.76	
	0.37	99.21	-167.4	8896.2		0.01	-0.50	-1.30	0.62	
	-0.49	98.58	-16.7	4856.8		-1.01	-0.96	0.09	0.06	
	-0.21	98.2	-199.9	2115.2		-0.68	-1.24	-1.60	-0.32	
Mean	0.359091	99.89727	-26.4273	4428.6						
SD	0.839958	1.372291	108.3649	7165.712						

Z-Score value is to understand **how far the data point is from the mean**. Technically, it measures the standard deviations below or above the mean.

Z-score normalization in data mining is useful for those kinds of data analysis wherein there is a need to compare a value with respect to a mean(average) value, such as results from tests or surveys.

For example, a person's weight is 150 pounds. Now, if there is a need to compare that value with the average weight of a population listed in a vast table of data, Z-score normalization is needed to study such values.

Method 2: Min-max normalization

What is easier to understand – the difference between 200 and 1000000 or the difference between 0.2 and 1?

Indeed, when the difference between the minimum and maximum values is less, the data becomes more readable.

The min-max normalization functions by converting a range of data into a scale that ranges from say for example 0 to 1.

- Min-max normalization: to $[new_min_A, new_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A$$

- Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0].
Then \$73,000 is mapped to

$$\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716$$

	X1	X2	X3	X4		X1	X2	X3	X4	
	-0.26	101.87	-9.2	19593.5		1.08	9.11	5.36	10.00	
	1.26	102.23	-71.7	3899.2		8.25	10.00	3.60	4.12	
	-0.47	99.78	113.9	-4608.1		0.09	3.92	8.81	0.93	
	1.61	101.07	41	888		9.91	7.12	6.77	2.99	
	-0.41	98.83	156.1	4810.4		0.38	1.56	10.00	4.46	
	1.63	99.92	-3.5	5490.1		10.00	4.27	5.52	4.71	
	0.87	98.67	-109.7	-7091.2		6.42	1.17	2.53	0.00	
	0.05	100.51	-23.6	9864.5		2.55	5.73	4.95	6.35	
	0.37	99.21	-167.4	8896.2		4.06	2.51	0.91	5.99	
	-0.49	98.58	-16.7	4856.8		0.00	0.94	5.15	4.48	
	-0.21	98.2	-199.9	2115.2		1.32	0.00	0.00	3.45	
min	-0.49	98.2	-199.9	-7091.2	Note: Max is 10, Min is 0					
max	1.63	102.23	156.1	19593.5						

Method 3: Decimal Scaling normalization

Decimal scaling is another technique for normalization in data mining. It functions by converting a number to a decimal point.

$$v' = \frac{v}{10^j}.$$

- V' is the new value after applying the decimal scaling
- V is the respective value of the attribute
- integer J defines the movement of decimal points.

Decimal scaling can tone down big numbers into easy to understand smaller decimal values

Suppose a company wants to compare the salaries of the new joiners. Here are the data values:

Name	Salary	Salary after Decimal Scaling
ABC	10,000	0.1
XYZ	25, 000	0.25
PQR	8, 000	0.08
MNO	15,000	0.15

$$v' = \frac{v}{10^j}.$$

How to use decimal scaling: look for the maximum value in the data. In this case, it is 25,000. Count the number of digits in this value. In this case, it is '5'. So here 'j' is equal to 5.

This means the V (value of the attribute) needs to be divided by 100,000 here.

Seatwork

Perform z-score normalization and min-max normalization (max is 1, min is 0) on the following data:

8

10

15

20