AI 330: Machine Learning

Fall 2023

Dr. Wessam EL-Behaidy

Associate Professor, Computer Science Department,

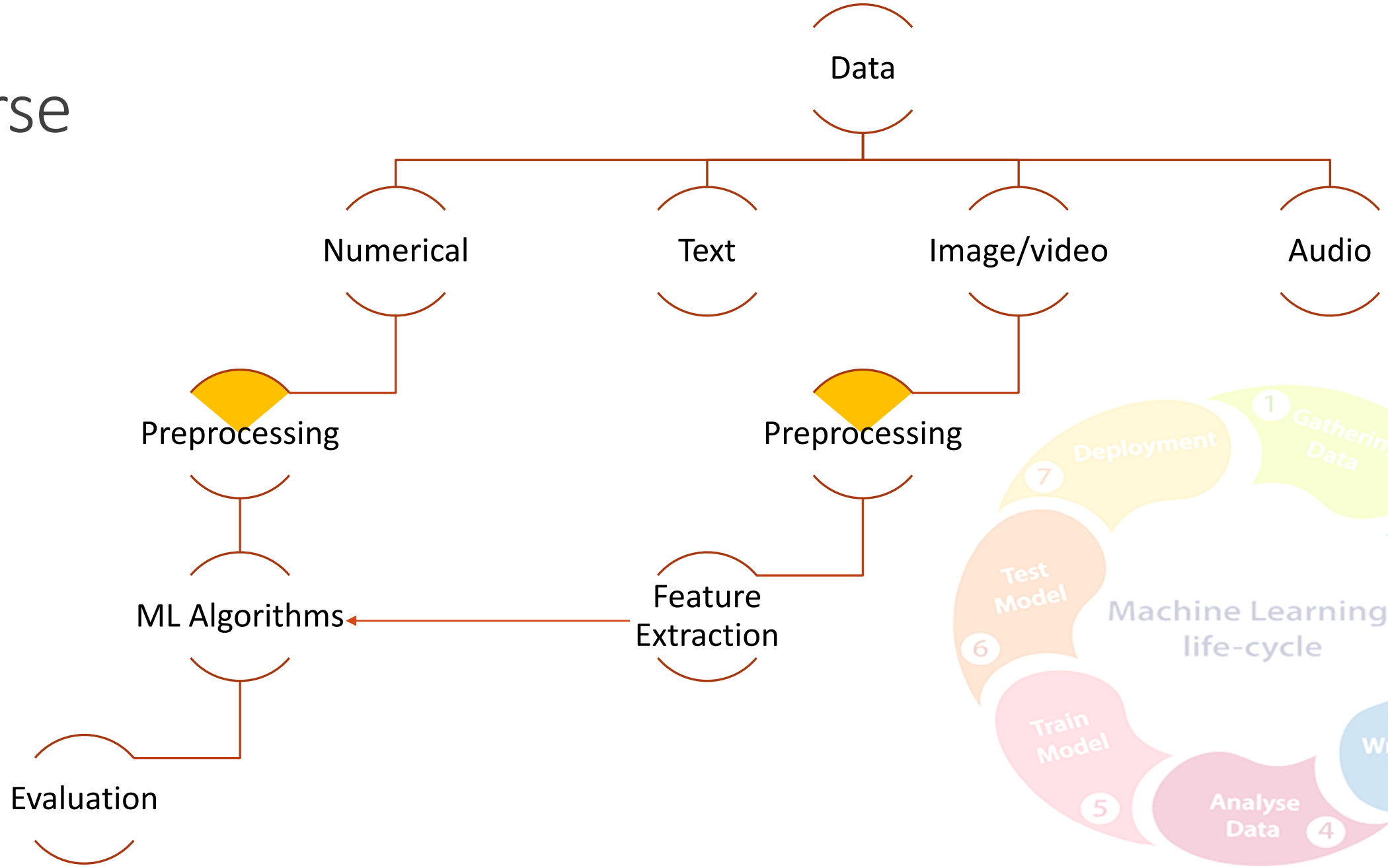
Faculty of Computers and Artificial Intelligence,

Helwan University.

Dr. Ensaf Hussein

Associate Professor, Computer Science Department,
Faculty of Computers and Artificial Intelligence,
Helwan University.

Course Map


Data
Numerical
Text
Image/video
Audio
Preprocessing
Preprocessing
ML Algorithms
Feature Extraction
Evaluation

1 Gathering Data
2 Data preparation
3 Data Wrangling
4 Analyse Data
5 Train Model
6 Test Model
7 Deployment
Machine Learning life-cycle

Lecture 2

Preprocessing Data

Numerical Data

Data Cleansing - Data Normalization

Image Preprocessing

Slides of:
CSE 412: SELECTED TOPICS IN COMPUTER ENGINEERING, Faculty of Engineering, Ain Shams University.

Numerical Data:
DATA CLEANSING

Reasons for Data Cleansing

- The data to be analyzed may be:
  - **Incomplete**; where the data is missing
    - **Filling-in Missing Values**
  - **Noisy**; where data may contain errors or outlier values
    - **Identifying and Removing Outliers**
    - **Smoothing Noisy Data**
  - **Inconsistent**; where data may contain discrepancies in the values
    - **Resolving Inconsistency**

Typical Example (Incomplete Data)

missing values ←

Field1 (Numeric)	Field2 (Catrgorical)	Field3 (Numeric)	Field4 (Numeric)
21	A	300	67
	A	250	
	B	280	93
24	W		76
22	C	500	85
12		350	66
11		330	
16	A	220	84
16	C		98
17		420	78
18	W	360	89

A set of fields with missing values

Reasons for Incomplete Data

- **Relevant data may not be recorded because:**
  - A misunderstanding from the data entry persons
  - Equipment failure
- **Relevant data may not be available because it is unknown or providing it is optional**

Dealing with Incomplete Data

There are several ways to deal with missing data:

- Replace the missing value with the field **mean** for the fields that take numerical values or the **mode** (if exists) for the fields that take categorical values
- Replace the missing values with a value generated at **random** from the field distribution observed
- Replace the missing value with some **default value**

Mean, Median, and Mode

- The mean for a population of size *n* can be computed by:

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$$

- Consider the following list of 9 numbers:

13, 15, 12, 17, 22, 11, 13, 19, 12

Mean = (**13 + 15 + 12 + 17 + 22 + 11 + 13 + 19 + 12**)/9 = **14.88889**

Mean, Median, and Mode

- The **median** is the middle value of the **ordered list of numbers**.
  - Consider the following list of **9** numbers:

13, 15, 12, 17, 22, 11, 13, 19, 12

- To compute the median, you need first to order the numbers:

11, 12, 12, 13, 13, 15, 17, 19, 22

- Hence, the median is **13**

Mean, Median, and Mode

- The **median** is the middle value of the **ordered list of numbers**.
  - Consider the following list of **10** numbers:

    13, 15, 12, 17, 22, 11, 13, 19, 12, 14

  - To compute the median, you need first to order the numbers:

    11, 12, 12, 13, 13, 14, 15, 17, 19, 22

  - Hence, the median is **(13 + 14)/2 = 13.5**

Mean, Median, and Mode

- The **mode** of a set of data is the value in the set that **occurs most often**.
  - Consider the following list of numbers:

13, 15, 12, 17, 22, 11, 13, 19, 13

Number	Occurrence
13	3
15	1
12	1
17	1

Number	Occurrence
22	1
11	1
19	1

Mode is **13**

Mean, Median, and Mode

- The **mode** of a set of data is the value in the set that **occurs most often**.
  - Consider the following list of numbers:

13, 15, 12, 17, 22, 11, 13, 19, 12

Number	Occurrence
13	2
15	1
12	2
17	1

Number	Occurrence
22	1
11	1
19	1

Mode is **13** and**12** **(Bimodal)**

Mean, Median, and Mode

- The **mode** of a set of data is the value in the set that **occurs most often**.
  - Consider the following list of numbers:

13, 15, 12, 17, 22, 11, 19

Number	Occurrence
13	1
15	1
12	1
17	1

Number	Occurrence
22	1
11	1
19	1

There is **no mode**

Handling Missing Values

Field1 (Numeric)	Field2 (Catrgorical)	Field3 (Numeric)	Field4 (Numeric)
21	A	300	67
17	A	250	85
17	B	280	93
24	W	334	76
22	C	500	85
12	A	350	66
11	A	330	85
16	A	220	84
16	C	334	98
17	A	420	78
18	W	360	89

A set of fields with missing values

Handling Missing Values (Using Means and Modes)

- Use the **mean** for the *numeric fields* and the **mode** (if exists) for the *categorical* fields
- **If mode doesn't exist,** you need to rely on either a default value or to use a random value

باحط عشوائي أو افتراضي

Handling Missing Values (Using Means and Modes)

In our example

- Numeric Fields: Field1, Field3, and Field4
  - Field1 Mean = (21+24+22+12+11+16+16+17+18)/9 = **17.44**
  - Field3 Mean = **334.44**
  - Field4 mean = **81.78**
- If any field doesn't accept decimal values, just approximate the mean value

Handling Missing Values (Using Means and Modes)

In our example

- Field2 is categorical, hence we need to compute the mode from the existing values

Category	Occurrence
A	3
B	1
W	2
C	2

- Hence, the mode is A

Handling Missing Values (Using Means and Modes)

Field1 (Numeric)	Field2 (Catrgorical)	Field3 (Numeric)	Field4 (Numeric)
21	A	300	67
17	A	250	82
17	B	280	93
24	W	334.44	76
22	C	500	85
12	A	350	66
11	A	330	82
16	A	220	84
16	C	334.44	98
17	A	420	78
18	W	360	89

Assumptions

- **Assume Field 1 and Field4 don't accept decimal numbers, Hence we approximate the mean**
- **Field3 accepts decimal numbers, hence we don't approximate the mean value**

Dealing with Incomplete Data

There are several ways to deal with missing data:

- Replace the missing value with the field **mean** for the fields that take numerical values or the **mode** (if exists) for the fields that take categorical values
- Replace the missing values with a value generated at **random** from the field distribution observed
- Replace the missing value with some **default value**

Handling Missing Values (Using Random Values)

Field1 (Numeric)	Field2 (Catrgorical)	Field3 (Numeric)	Field4 (Numeric)
21	A	300	67
22	A	250	73
17	B	280	93
24	W	359.97	76
22	C	500	85
12	C	350	66
11	A	330	86
16	A	220	84
16	C	327.01	98
17	W	420	78
18	W	360	89

Assumption

- **Field 2 has no mode, Hence we use the random values**

Dealing with Incomplete Data

There are several ways to deal with missing data:

- Replace the missing value with the field **mean** for the fields that take numerical values or the **mode** (if exists) for the fields that take categorical values
- Replace the missing values with a value generated at **random** from the field distribution observed
- Replace the missing value with some default value

Handling Missing Values (Using Default Values)

Field1 (Numeric)	Field2 (Catrgorical)	Field3 (Numeric)	Field4 (Numeric)
21	A	300	67
0	A	250	50
0	B	280	93
24	W	240	76
22	C	500	85
12	N	350	66
11	N	330	50
16	A	220	84
16	C	240	98
17	N	420	78
18	W	360	89

Defaults

- **Field1 Default: 0**
- **Field3 Default: 240**

Field2 Default: N
Field4 Default: 50

Reasons for Data Cleansing

- The data to be analyzed may be:
  - **Incomplete**; where the data is missing
    - **Filling-in Missing Values**
  - **Noisy**; where data may contain errors or outlier values
    - **Identifying and Removing Outliers**
    - **Smoothing Noisy Data**
  - **Inconsistent**; where data may contain discrepancies in the values
    - **Resolving Inconsistency**

Typical Example (Outlier Values)

Customer ID	Zip	Gender	Income	Age	Marital Status	Transaction Amount
1001	10048	M	75000	C	M	5000
1002	J2S7K7	F	−40000	40	W	4000
1003	90210		10000000	45	S	7000
1004	6269	M	50000	0	S	1000
1005	55101	F	99999	30	D	3000

Sample Table

Reasons:

- Outliers are data values that deviate from expected values of the rest of the data set
- The values 10000000 and -40000 look very divergent from the rest of values

Handling Outliers

Customer ID	Zip	Gender	Income	Age	Marital Status	Transaction Amount
1001	10048	M	75000	C	M	5000
1002	J2S7K7	F	–40000	40	W	4000
1003	90210		10000000	45	S	7000
1004	6269	M	50000	0	S	1000
1005	55101	F	99999	30	D	3000

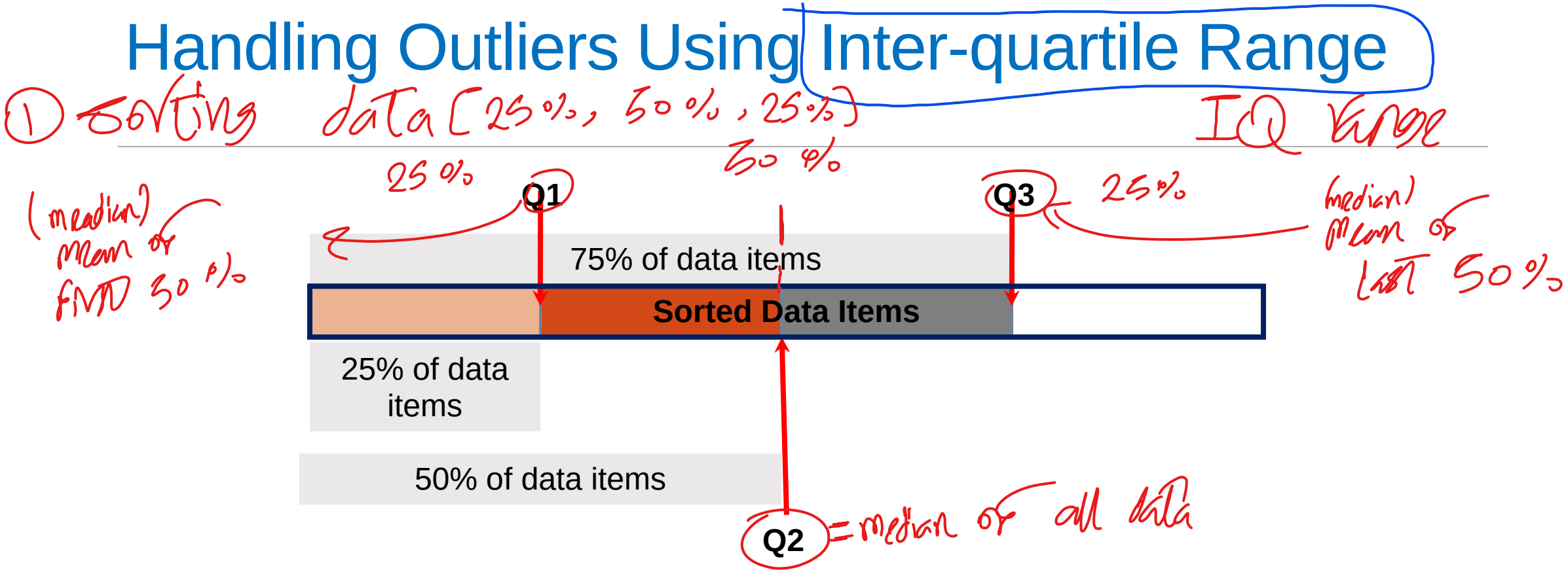
Sample Table

Data Set

Possible Outlier Values

- Outliers are data values that deviate from expected values of the rest of the data set
- Outliers are extreme values that lie near the limits of the data range or go against the trend of the remaining data.
- Normally, outliers need **more investigation** to make sure that they don't occur due to mistakes during data entry

Handling Outliers Using Inter-quartile Range



- **Quartile is any of the three values which divide the sorted data set into four equal parts**
- **First quartile (Q1) cuts off lowest 25% of data**
- **Second quartile (Q2) cuts data set in half (it is the median of the data set)**
- **Third quartile (Q3) cuts off highest 25% of data, or lowest 75%**

Computing Q1, Q2, and Q3

- to compute **Q1**, **Q2**, and **Q3**, use the following method:
  - Order the given data set in ascending order.
  - Use the median to divide the ordered data set into two halves. This median is second quartile (Q2). **Exclude this median (if it is one of the data items)** from any further computation.
  - The first quartile (Q1) value is the median of the lower half of the data.
  - The third quartile (Q3) value is the median of the upper half of the data.

Example #1 of Computing Q1, Q2, and Q3

- Compute Q1, Q2, and Q3 for the following data set:

  6, 47, 49, 15, 42, 41, 7, 39, 43, 40, 36

  - Order the given data set in ascending order:

    6, 7, 15, 36, 39, 40, 41, 42, 43, 47, 49

  - Q2 = 40 (median of the data set)
  - Q1 is the median of the lower half of the data (shown in red).
  - Q1 = 15
  - Q3 is the median of the upper half of the data (shown in green).
  - Q3 = 43

Example #2 of Computing Q1, Q2, and Q3

- Compute Q1, Q2, and Q3 for the following data set:

39, 36, 7, 40, 41, 17

  - Order the given data set in ascending order:

7, 17, 36, 39, 40, 41

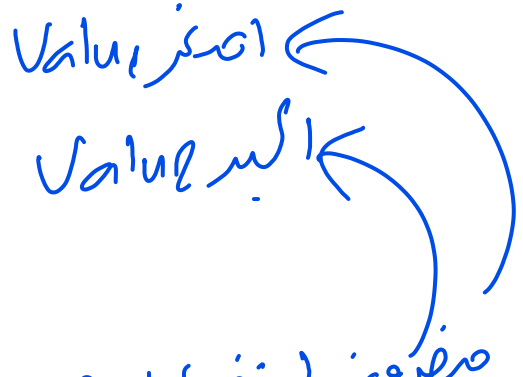
  - Q2 = (36+39)/2 = 37.5 (median of the data set). Note, the number of data items is even so the median is the average of the middle two data items
  - The median is not a data item, hence we need to use all items in the first half of the data items to compute Q1 and the rest of the items are used to compute Q3
  - Q1 = 17
  - Q3 is the median of the upper half of the data (shown in green).
  - Q3 = 40

Detecting Outliers using Inter-quartile Range

- Compute the Inter-Quartile Range (**IQR**) as follows:

$$IQR = Q3 - Q1$$

- A data value is an outlier if:
  - its value is <= (Q1 – 1.5*IQR), or
  - its value is >= (Q3 + 1.5*IQR).



Example of Detecting Outliers using Inter-quartile Range

Customer ID	Zip	Gender	Income	Age	Marital Status	Transaction Amount
1001	10048	M	75000	C	M	5000
1002	J2S7K7	F	−40000	40	W	4000
1003	90210		10000000	45	S	7000
1004	6269	M	50000	0	S	1000
1005	55101	F	99999	30	D	3000

Sample Table

Data Set that might contains outliers

- Data Set

  75000, -40000, 10000000, 50000, 99999

Example of Detecting Outliers using Inter-quartile Range

- Data Set:

  75000, -40000, 10000000, 50000, 99999

- Ordered Data Set:

  -40000, 50000, 75000, 99999, 10000000

- **Q2 =** 75000, **Q1 =** (–40000+50000)/2 = 5000, **Q3 =** (99999+10000000)/2 = 5049999.5
- IRQ = Q3 – Q1
  = 5049999.5 – 5000
  = 5044999.5
- Q1 – 1.5*IRQ = 5000 – 1.5*50449999.5 = – 7562499.5
- Q3 + 1.5*IRQ = 5049999.5 + 1.5*5044999.5 = 12617498.75
- All data in the data set are within range, hence there is no outliers in this example

Example of Detecting Outliers using Inter-quartile Range

Customer ID	Zip	Gender	Income	Age	Marital Status	Transaction Amount
1001	10048	M	75000	C	M	5000
1002	J2S7K7	F	40000	40	W	4000
1003	90210		10000000	45	S	7000
1004	6269	M	50000	0	S	1000
1005	55101	F	99999	30	D	3000
1006	55101	F	75000	30	M	3000

Data Set that might contains outliers

- Data Set

  75000, 40000, 10000000, 50000, 99999, 75000

Example of Detecting Outliers using Inter-quartile Range

- Data Set:

  75000, 40000, 10000000, 50000, 99999, 75000

- Ordered Data Set:

  40000, 50000, 75000, 75000, 99999, 10000000

- Q2 = (75000+ 75000)/2 = 75000, Q1 = 50000, Q3 = 99999
- IRQ = Q3 – Q1
  = 99999 – 50000
  = 49999
- Q1 – 1.5*IRQ = 50000 – 1.5*49999 = –24998.5
- Q3 + 1.5*IRQ = 99999 + 1.5* 49999 = 174997.5
- Hence data item **10000000** is an outlier and should be **re-investigated for any data-entry errors**

Typical Example (Noisy Data)

Customer ID	Zip	Gender	Income	Age	Marital Status	Transaction Amount
1001	10048	M	75000	C	M	5000
1002	J2S7K7	F	−40000	40	W	4000
1003	90210		10000000	45	S	7000
1004	6269	M	50000	0	S	1000
1005	55101	F	99999	30	D	3000

Sample Table

Reasons:

- The Zip code consists of five digits and cannot contain any letters
- Income must be positive number
- Age must be positive number

Noisy Data

- Noisy data are the kind of data that have incorrect values
- Some reasons for noisy data:
  - Data collection instruments may be faulty
  - Human or computer errors may occur during data entry
  - Transmission errors may occur
  - Technology limitations like buffer size, may occur during data-entry

Smoothing Noisy Data

- By **smoothing noisy data** we can correct the errors
- Smoothing noisy data is performed by:
  1. Validation and correction
  2. Standardization

Validation and Correction of Noisy Data

- This step examines the data for data-entry errors and tries to correct them automatically as far as possible according to the following guidelines:
  - **Spell checking** based on dictionary lookup is useful for identifying and correcting misspellings.

    Example: **Kairo** can be spell-checked and corrected into **Cairo**

  - Use **dictionaries on geographic names** and **zip codes** helps to correct address data.

    Example: Zip code **1243456** can be detected as an error since there is no Zip code matches this value

Validation and Correction of Noisy Data *(Cont.)*

- Check validation rules and make sure field values follow the rules; for example:
  - Age is not less than certain amount and age is a positive number.

    Example: if there is a rule governing your data says that age must be between 20 and 60, then ages of 18, 15, and 68 are detected as errors
  - Each value of the categorical values belong to certain category.

    Example: if all the categories you have are A, B, C, and D, Then if categories W or N are found they will be declared as errors

Validation and Correction of Noisy Data *(Cont.)*

- Check the fields that have ambiguous values and check for any possible data-entry errors

Example: Using the same category value to refer to different meaning. "**S**" in "Marital Status" field could refer to "**S**ingle" or "**S**eparated"

Typical Example (Ambiguity)

Customer ID	Zip	Gender	Income	Age	Marital Status	Transaction Amount
1001	10048	M	75000	C	M	5000
1002	J2S7K7	F	–40000	40	W	4000
1003	90210		10000000	45	S	7000
1004	6269	M	50000	0	S	1000
1005	55101	F	99999	30	D	3000

Sample Table

Reasons:

- "**S**" in Marital Status could refer to "**S**ingle" or "**S**eparated"
- So, there is a kind of **ambiguity** in the data

Standardization to Smooth Noisy Data

- Data values should be consistent and have a uniform format. For example:
  - Date and time entries should have a specific format

Oct. 19, 2009 **10/19/2009** **19/10/2009**

All dates must be written in the same format
that have been agreed upon (e.g., Day/Month/Year)

  - Names and other string data should be converted to either upper or lower case.

MOHAMED AHMED **instead of** **Mohamed Ahmed**

  - Removing prefixes and suffixes from names.

Mohamed Ahmed instead of Mr. Mohamed Ahmed
Mohamed Ahmed instead of Mohamed Ahmed, Ph.D.

Standardization to Smooth Noisy Data *(Cont.)*

- Abbreviations and encoding schemes should consistently be resolved by consulting special dictionaries or applying predefined conversion rules.

US is the standard abbreviation of **United States**

Reasons for Data Cleansing

- The data to be analyzed may be:
  - **Incomplete**; where the data is missing
    - **Filling-in Missing Values**
  - **Noisy**; where data may contain errors or outlier values
    - **Identifying and Removing Outliers**
    - **Smoothing Noisy Data**
  - **Inconsistent**; where data may contain discrepancies in the values
    - **Resolving Inconsistency**

تناقضات

Data Inconsistency

- **Data inconsistency** means that different data items contain discrepancies in their values
- It can occur when different data items depend on other data items and their values don't match; for example:
  - **Age** and **Birth-date**; age can be computed from the birth-date, hence the value of Age must match the value computed from the birth-date
  - **City** and **Phone-area-code**; each city has certain area-code
  - **Total-price** and (**unit-price** and **quantity**); total-price can be computed from the unit-price and quantity
- These dependencies can be utilized to detect errors and substitute missing values or correct wrong values

Data Inconsistency Example

Client_Code	Place	Phone	Quantity	Unit_Price	Total_Price
101	Cairo	02-99xxxxxx	10	120	1200
102	Cairo	03-99xxxxxx	10	234	2340
102	Alexadria	03-99xxxxxx	20	400	600
104	Port-Said	03-99xxxxxx	30	100	300

Example of Inconsistent Data

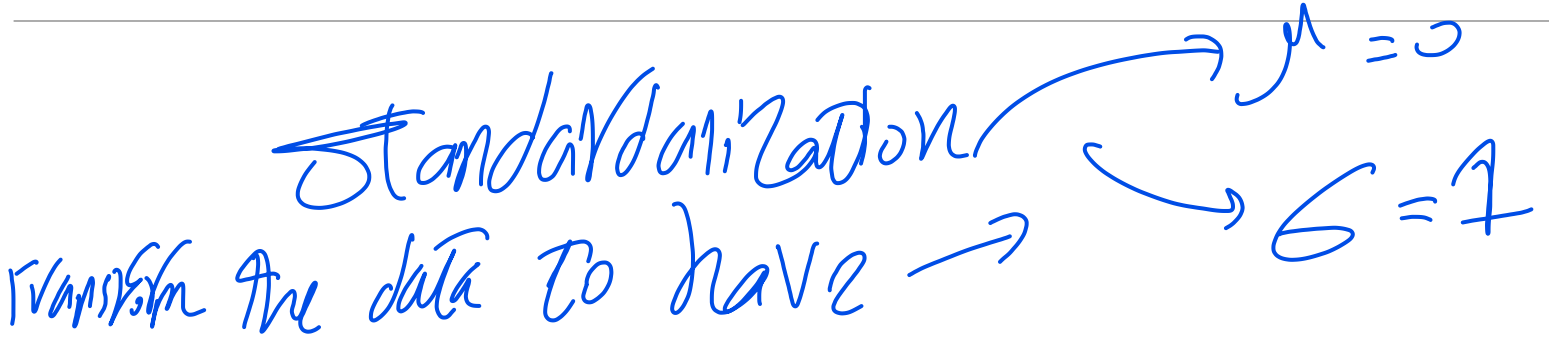
Client_Code	Place	Phone	Quantity	Unit_Price	Total_Price
101	Cairo	02-99xxxxxx	10	120	1200
102	Cairo	03-99xxxxxx	10	234	2340
102	Alexadria	03-99xxxxxx	20	400	600
104	Port-Said	03-99xxxxxx	30	100	300

Data Inconsistency Marked in Red

Incorrect Area-Code

Total_Price doesn't equal (Quantity*Unit_Price)

Numerical Data:
DATA NORMALIZATION



Data Normalization

- Data Normalization means that the attribute data are scaled to fall within a small specified range such as –1.0 to +1.0, or 0.0 to 1.0
- Normalization is important for data classification and analysis by data mining techniques

Min-Max Normalization

- Min-Max normalization performs a linear transformation on the original data
- Suppose that min_A and max_A are the minimum and maximum values of an attribute A
- Min-Max normalization maps any value v of A to v' in the range[new_min_A, new_max_A] by computing:

$$v' = \left(\left[\frac{(v - min_A)}{(max_A - min_A)} \right] \times \left[(new\_max_A - new\_min_A) \right] \right) + new\_min_A$$

- Min-max normalization preserves the relationships among the original data values

Min-Max Normalization: Example

- Let the minimum and maximum values for the attribute income be $12000 and $98000
- To map income to the range [0.0, 1.0], we have:

min_A = 12000 **max_A = 98000**

new_min_A = 0.0 **new_max_A = 1.0**

- By min-max normalization, a value of $73600 for income is transformed to:

$$\left(\frac{73600-12000}{98000-12000}\right)\times[1.0-0.0]+0.0=0.716$$

Mean Normalization

- In mean normalization, the values for attribute A are centralized at zero and rescaled on the range of values of A.
- A value v of A is normalized to v' by computing:

$$v' = \frac{v - \bar{A}}{max_A - min_A}$$

Where $\bar{A} = \left(\frac{\sum_{i=1}^{n} A}{n} \right)$

- Where $\bar{A}$ is the mean value of A and min_A and max_A are the minimum and maximum values of an attribute A

Mean Normalization: Example

- Let the minimum and maximum values for the attribute "income" be $12000 and $98000, with an average $ 54000
- So, we have:

  $\bar{A}$ **= 54000** **$\min_A$ = 12000** **$\max_A$ = 98000**
- By mean normalization, a value of $73600 for income is transformed to:

$$\frac{73600 - 54000}{98000 - 12000} = 0.228$$

Z-Score Normalization

- In Z-Score normalization, the values for an attribute A are normalized based on the mean and standard deviation of A
- A value v of A is normalized to v' by computing:

$$v' = \left(\frac{v - \overline{A}}{\sigma_A} \right)$$

Where $\overline{A} = \left(\frac{\sum_{i=1}^{n} A}{n} \right)$ and $\sigma_A = \sqrt{\frac{\sum_{i=1}^{n} (A_i - \overline{A})^2}{n-1}}$

- Where $\overline{A}$ is the mean value of A and σ_A is the standard deviation of A
- The Z-Score normalization method is useful when:
  - the actual minimum and maximum of attribute A are **unknown**
  - there are **outliers** that dominate the min-max normalization

Z-Score Normalization: Example

- Suppose that the mean and standard deviation of the values for the attribute income are $54000 and $16000, respectively
- With Z-Score normalization, a value of $73600 for income is transformed to

$$\frac{73600-54000}{16000}=1.225$$

IMAGE PREPROCESSING

Image Preprocessing Aim

The aim of pre-processing is an improvement of the image data that suppresses **undesired distortions** or **enhances** some image **features** relevant for further processing and analysis tasks.

Image Preprocessing Techniques

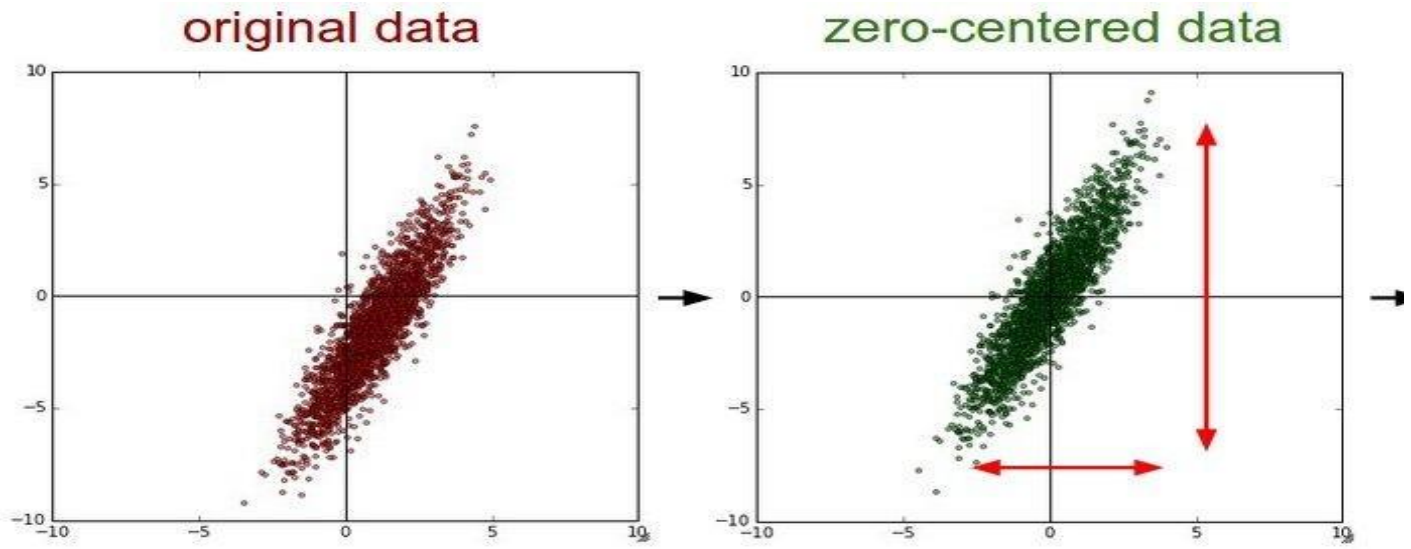
There are 4 different types of Image Pre-Processing techniques:

1) Pixel brightness transformations/corrections
   - Histogram equalization, etc.
2) Geometric Transformations **(Data Augmentation)**
   - Scaling, rotation, translation, shearing, etc.
3) Image Filtering and Segmentation
   - Low-pass filtering (Smoothing), high-pass filters (Edge Detection, Sharpening) , etc.
   - Colour thresholding, Region similarity, etc.
4) Fourier transform and Image restoration

https://www.mygreatlearning.com/blog/introduction-to-image-pre-processing/?highlight=Image

Zero-Centered Data

- Subtract the mean image (mean image = [32,32,3] array)
- Subtract per-channel mean (mean along each channel = 3 numbers)



Convolutional Neural Networks for Visual Recognition (CS231n):
http://cs231n.stanford.edu/index.html

Thanks