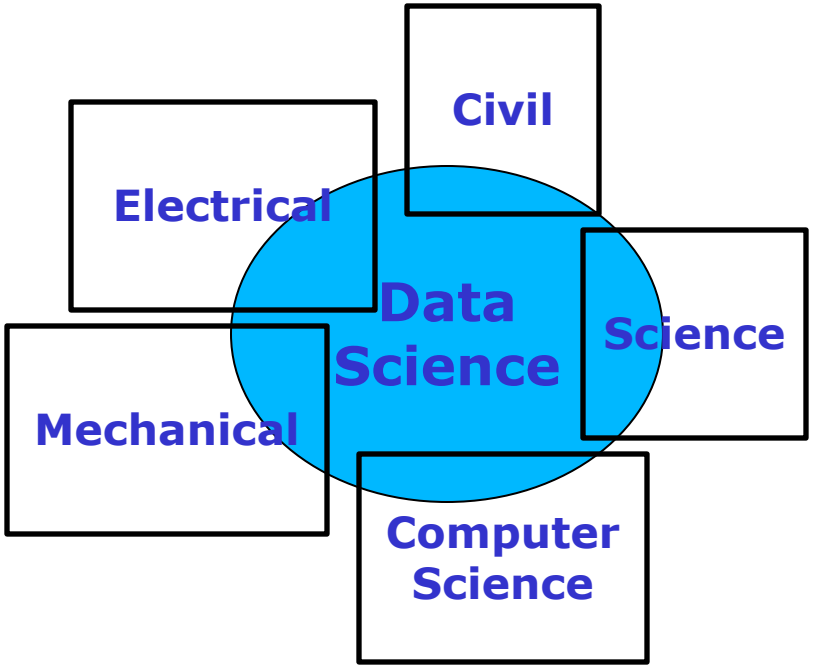
Data Science – III
(Data Processing and Modeling)

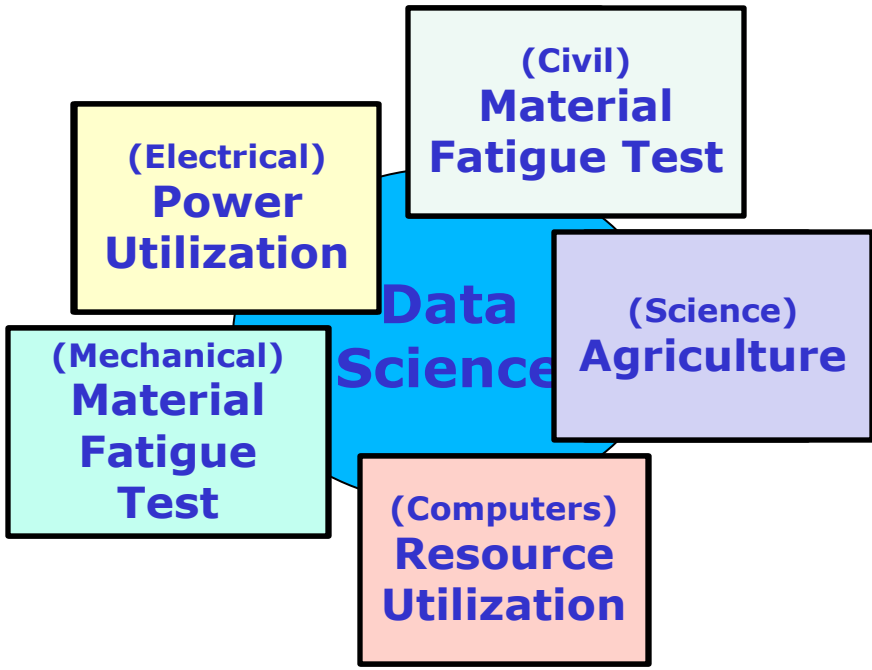
Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Data is available in every discipline



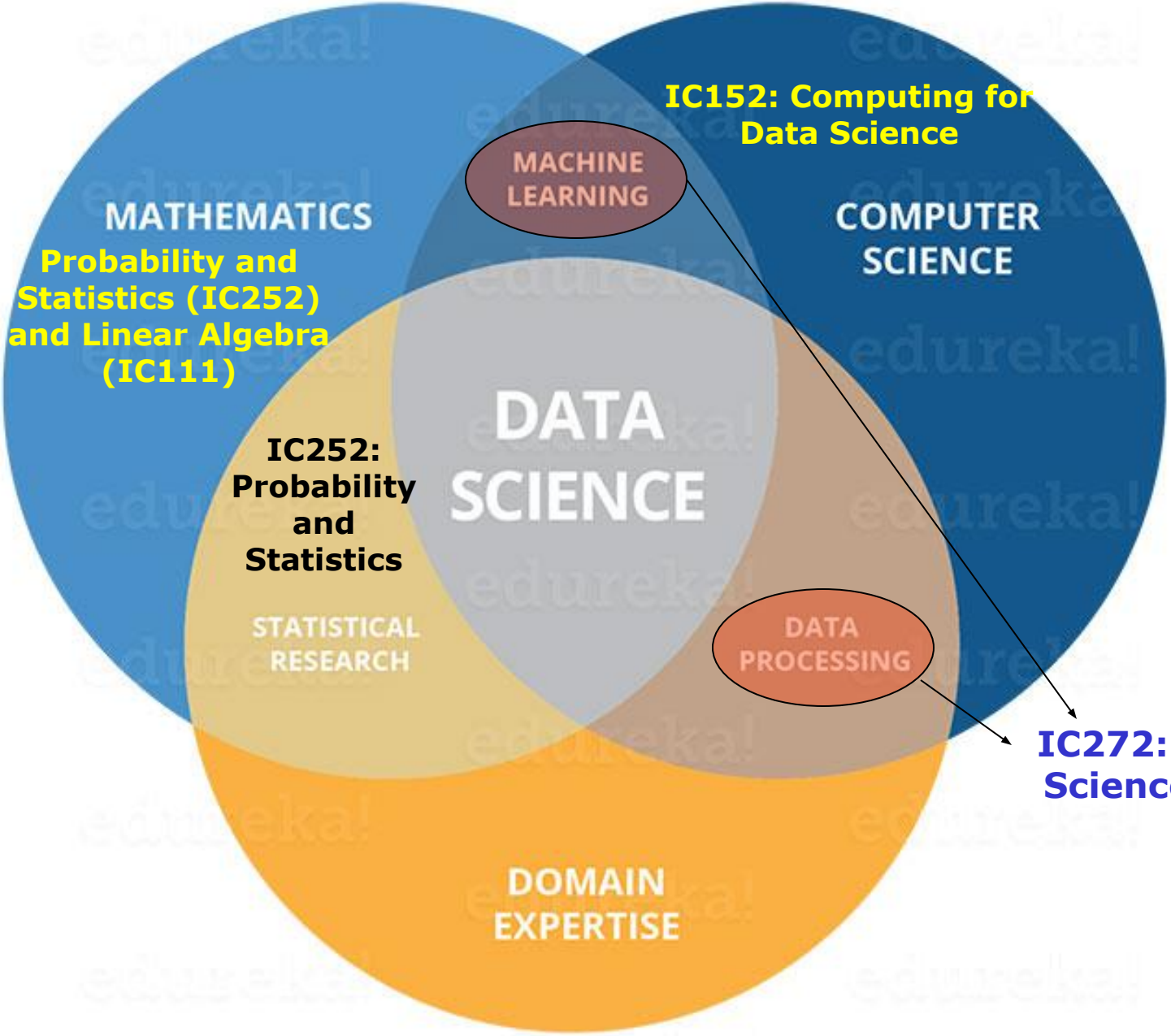
Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Data is available in every discipline

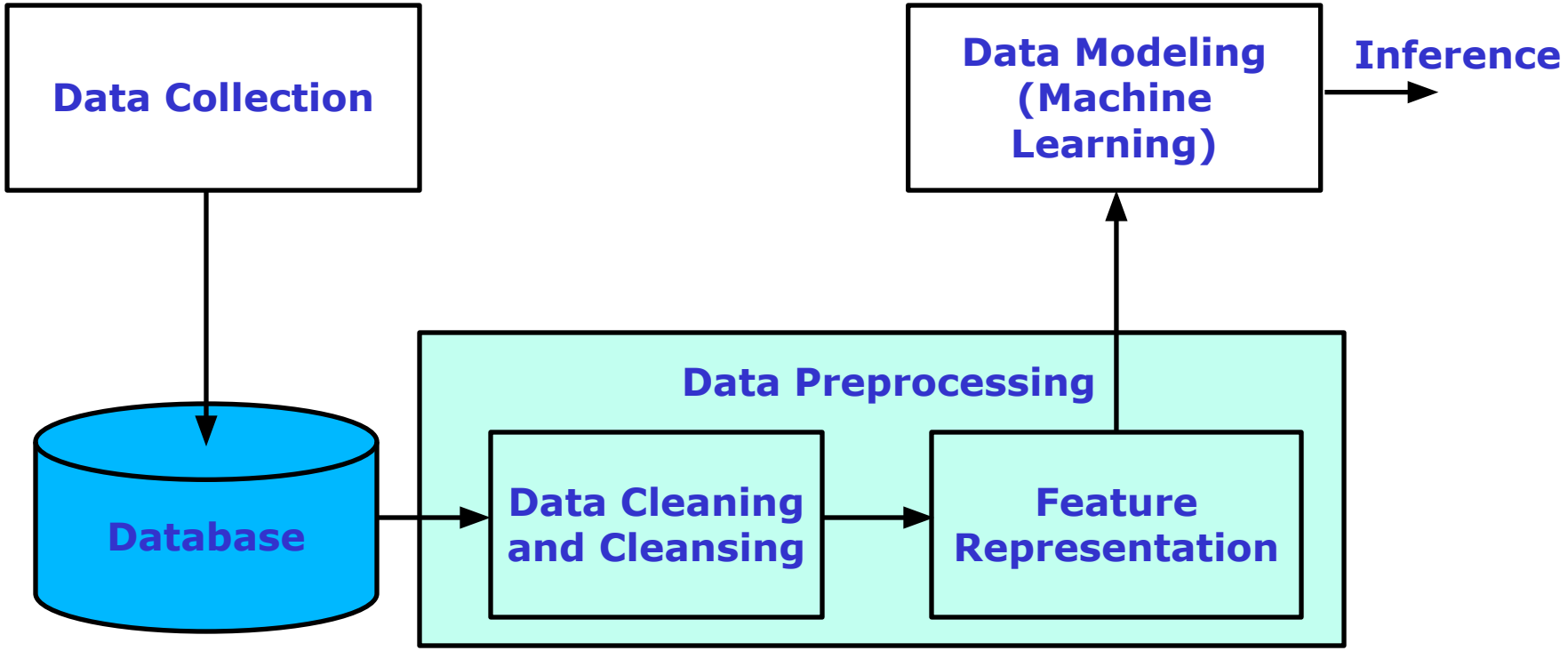


- Concept to unify statistics, data analysis, machine learning and their related methods to understand and analyse the actual phenomena with data
  - Includes theories from mathematics, statistics, computer science and domain from where the data is generated

Data Science



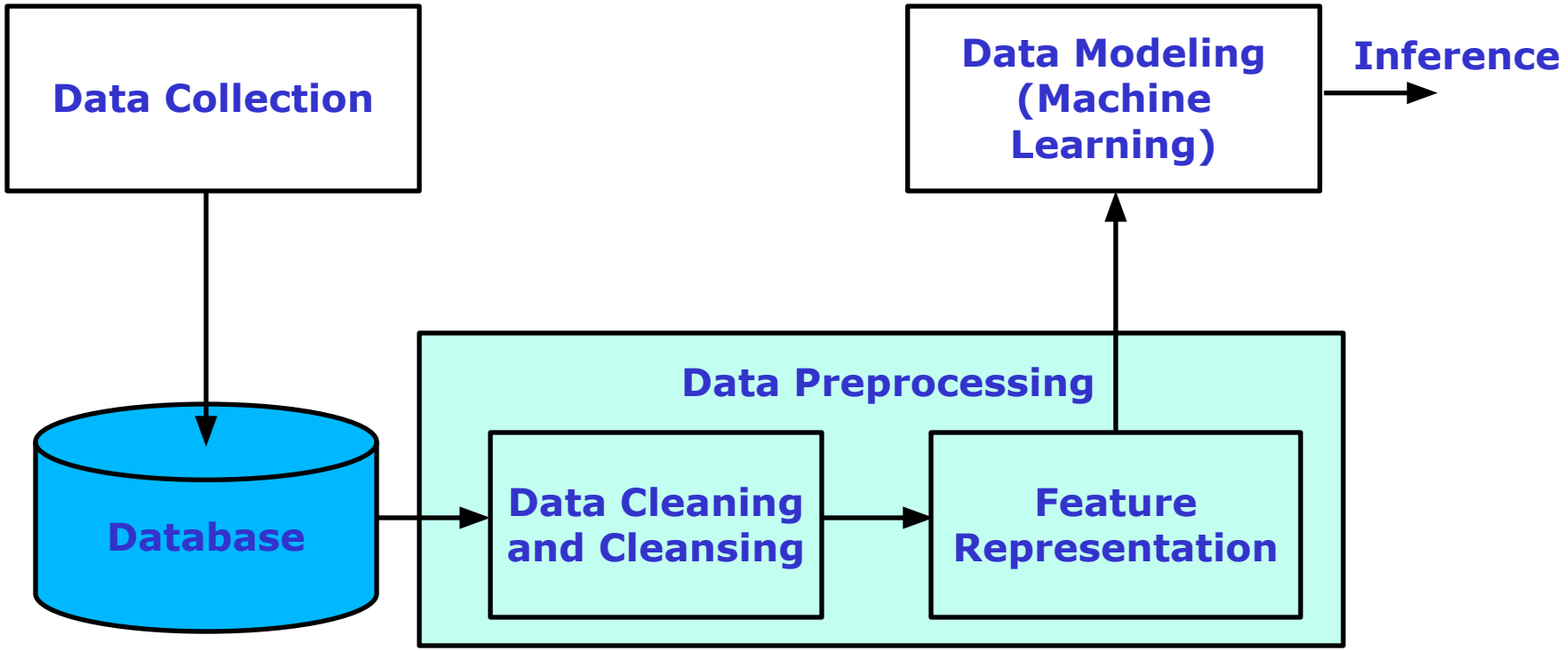
Data Science – III: Big Picture



Data, Types of Data and Data Collection using Sensors

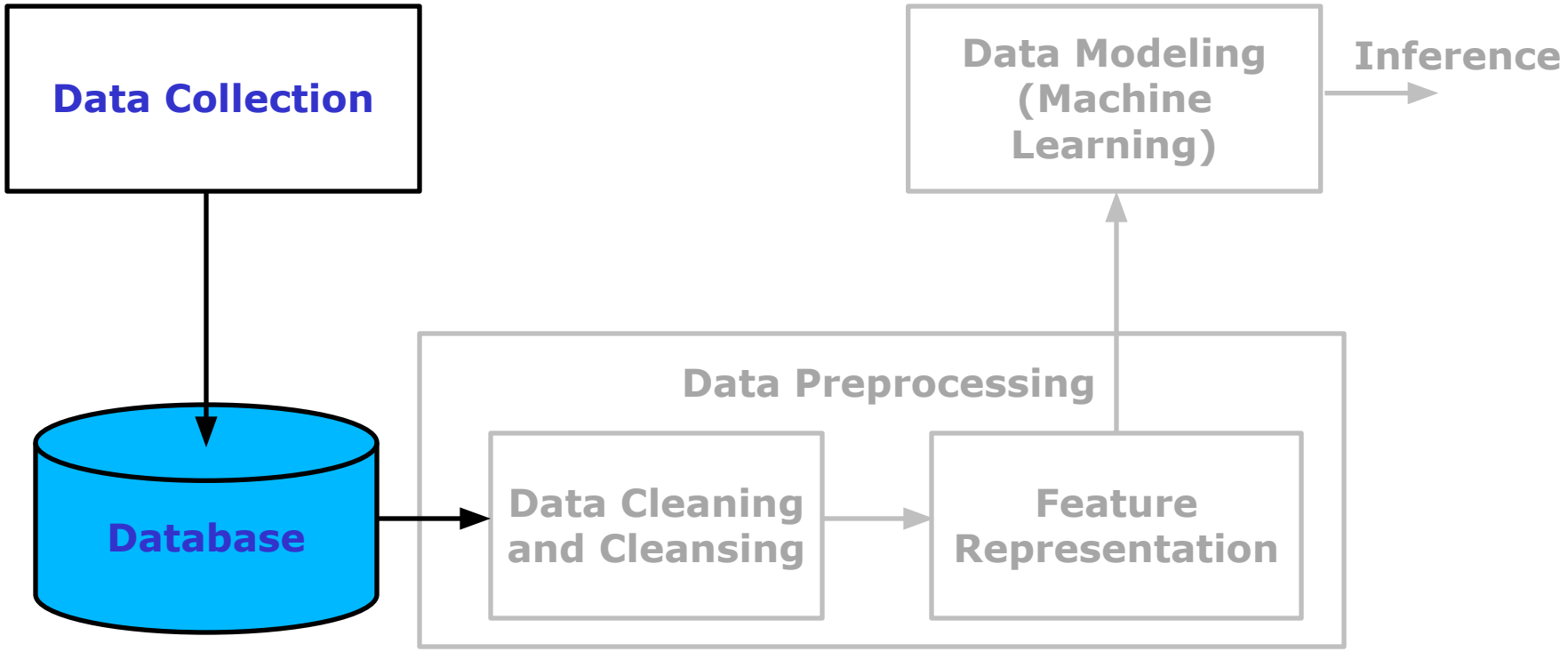
Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data
- Machine learning uses data to extract knowledge



Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data
- Machine learning uses data to extract knowledge



Data Collection

- Data manifests itself in many different forms
- Different forms of data require different ways to collect them and different storage solutions
- Collection of data may consists of sending out surveys, polls or doing other experiments
- Data based on the way it is collected:
  - Data that comes from surveys
    - Usually textual form of data or mixed

Sl. No	Timestamp	Email Address	Please mention your name	Please mention your Roll Number	Do you have a Laptop/Tablet which can help you to view videos?	What type of internet connection do you have?	Do you have the Laptop/Desktop for doing programming assignments?
78	9/14/2020 19:14:17	b19042@students.iitmandi.ac.in	Manan Shah	19042	Yes	4G and above	Yes
121	9/14/2020 20:31:32	b19107@students.iitmandi.ac.in	Rishabh Garg	19107	Yes	3G	Yes
25	9/14/2020 18:52:14	b19001@students.iitmandi.ac.in	Aarushi Gajri	B19001	Yes	4G and above	Yes
120	9/14/2020 20:30:22	b19002@students.iitmandi.ac.in	Aditya Narayan khokhar	B19002	Yes	3G	Yes
106	9/14/2020 19:56:31	b19003@students.iitmandi.ac.in	Aditya Sarkar	B19003	Yes	4G and above	Yes
146	9/14/2020 21:58:00	b19004@students.iitmandi.ac.in	Anooshka Bajaj	b19004	Yes	4G and above	Yes
4	9/15/2020 11:12:56	b19005@students.iitmandi.ac.in	Krishna sai	B19005	No	3G	No
83	9/14/2020 19:18:31	b19006@students.iitmandi.ac.in	Chirag	B19006	Yes	Bad internet connection fit only to view/download to lecture pdf files.	Yes
175	9/15/2020 8:30:34	b19008@students.iitmandi.ac.in	Samvivek	B19008	Yes	3G	Yes
215	9/16/2020 3:08:13	b19010@students.iitmandi.ac.in	Kshitij Nair	b19010	Yes	4G and above	Yes
69	9/14/2020 19:07:44	b19011@students.iitmandi.ac.in	Laishram	B19011	Yes	3G	Yes

Data Collection

- Data manifests itself in many different forms
- Different forms of data require different ways to collect them and different storage solutions
- Collection of data may consists of sending out surveys, polls or doing other experiments
- Data based on the way it is collected:
  - Data that comes from surveys
    - Usually textual form of data or mixed
  - Data entered in a database as system entry
    - E.g. Student information entered on academic automation system etc.
  - Data in the form of signals (comes from sensors)
    - Speech/Audio, Images and videos, Temperature readings, Humidity, Seismic data, EEG (all bio-type signals) etc.
- According to the objective of the task, the way the data is collected will change

Types of Data: Based on Organization

1. Unstructured data:
   - Rawest form of data
   - Example: Any type of files like texts, images, sounds or videos etc.
   - This type of data stored in a repository of files
     - Well organised directories on the computer hard drive

Types of Data: Based on Organization

2. Structured data:
   - It is a tabular data (rows and columns), which are very well defined

Date/ Time	Temperature (C)/ Humidity (%)	Pressure (Pa)	Rain (inches)	Light intensity (lux)	Accelerations (g)	Force (N)	Moisture (%)
2017-09-06 18:44:32	23.00,56.00	617.64	0.01	3	0.52,0.31,-0.80,0.00,0.00,0.00,31.36,-159.01	0.02	81.00
2017-09-06 18:33:32	24.00,58.00	619.47	0.01	12	0.52,0.30,-0.79,0.00,0.00,0.00,31.45,-159.12	0.02	82.00
2017-09-06 18:22:39	24.00,58.00	623.37	0.00	71	0.52,0.31,-0.80,0.00,0.00,0.00,31.35,-158.88	0.02	83.00
2017-09-06 18:11:31	25.00,60.00	627.02	0.05	194	0.51,0.31,-0.80,0.00,0.00,0.00,30.80,-159.00	0.02	81.00

   - Stored in databases
     - Spreadsheets [Comma Separated Value (CSV) format]
     - Oracle
     - DB2
     - MySQL etc.

Types of Data: Based on Organization

3. Semi-Structured data:
   - Anywhere between unstructured and structured data
   - A consistent format is defined, however there is no strict structure and parts of data may be incomplete or different type
   - Example: Data in the form of XML and JSON
     - Stored in document oriented databases

Types of Data: Based on Organization

3. Semi-Structured data:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>

  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>

  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>

  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
```

- Anywhere between unstructured and structured data
- A consistent format is defined, however there is no strict structure and parts of data may be incomplete or different type
- Example: Data in the form of XML and JSON
  - Stored in document oriented databases

Type of Data: Based on Variables (Value) found in Data

- Mainly in Structured Data:

1. Numerical data:
   - Data represented as numbers
   - Data in which information is measurable
   - This type of data is called quantitative data as its describes a quantity
   - Two types based on the values taken:
     - Continuous valued data:
       - Numbers does not have logical end
       - Range lies in the natural limit of what we are measuring
       - Example: Cost of the books, atmospheric temperature etc.
     - Discrete valued data:
       - Numbers have logical end
       - There is a specific limit on the range of the values
       - Example: number of members of family, number of days in a month, number of colours in flag etc.

Type of Data: Based on Variables (Value) found in Data

2. Categorical data:
   - Data that is not a number. It can be string of text or date
   - It describe an item or event to one of few different categories
   - Example: Ethnicity, gender, eye colour, etc.
   - This type of data is called qualitative data as its describes a quality
   - Three types values they hold:
     - Ordinal values: Values that have a set order to them
       - Example: Severity of a alarm as "Critical", "Medium" and ""Low", Ranking of a running race as " First", Second", Third"
     - Nominal values: Values that have no set order to them
       - Example: Values for the variables "Marital Status", "Country", "Eye Colour" etc.
     - Binary values: Special type of categorical data
       - Have only two values – "Yes" and "No" OR "True" and "False" OR "1" and "0"

Type of Data: Based on Variables (Value) found in Data

3. Time series data:
   - Series of data. It involve time and some kind of value
   - Example: Temperature at every hour
   - It is clearly structured and numeric in nature
   - Special case of numerical data
   - This type of data is important because of IoT and sensors
   - Data from sensors are almost always time-series in nature

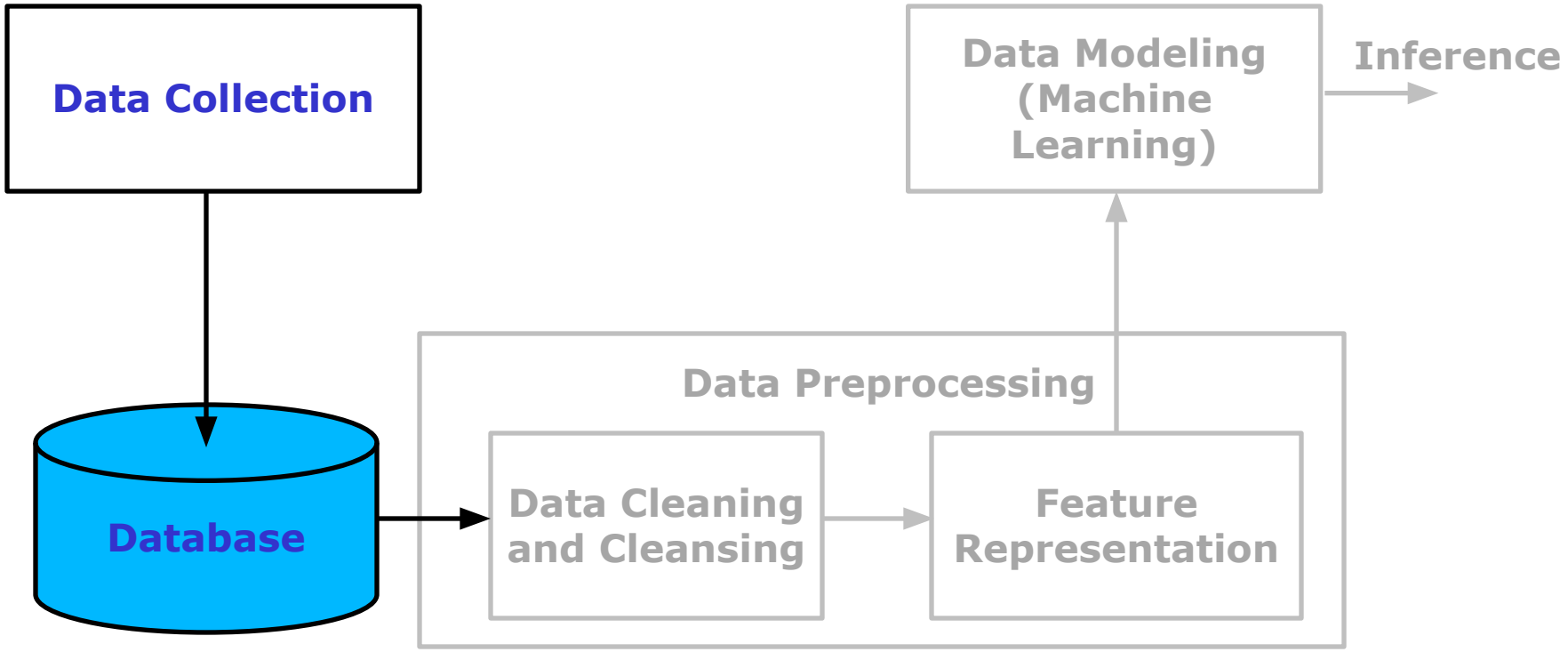
Date/ Time	Temperature (C)/ Humidity (%)	Pressure (Pa)	Rain (inches)	Light intensity (lux)	Accelerations (g)	Force (N)	Moisture (%)
2017-09-06 18:44:32	23.00,56.00	617.64	0.01	3	0.52,0.31,-0.80,0.00,0.00,0.00,31.36,-159.01	0.02	81.00
2017-09-06 18:33:32	24.00,58.00	619.47	0.01	12	0.52,0.30,-0.79,0.00,0.00,0.00,31.45,-159.12	0.02	82.00
2017-09-06 18:22:39	24.00,58.00	623.37	0.00	71	0.52,0.31,-0.80,0.00,0.00,0.00,31.35,-158.88	0.02	83.00
2017-09-06 18:11:31	25.00,60.00	627.02	0.05	194	0.51,0.31,-0.80,0.00,0.00,0.00,30.80,-159.00	0.02	81.00

Data, Types of Data and Data Collection using Sensors

Need for Data Preprocessing

Summary of Previous Class:

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data
- Machine learning uses data to extract knowledge



Summary of Previous Class:

Types of Data: Based on Organization

1. Unstructured data:
2. Structured data:
   - It is a tabular data (rows and columns), which are very well defined
   - Each row is finite ordered list (sequence) of elements, where each element in a column is belonging to an attribute of specific type
   - Example: Spreadsheets [Comma Separated Value (CSV) format]
3. Semi-structured data:

Summary of Previous Class:

Type of Data: Based on Variables (Value) found in Data

- Mainly in Structured Data:

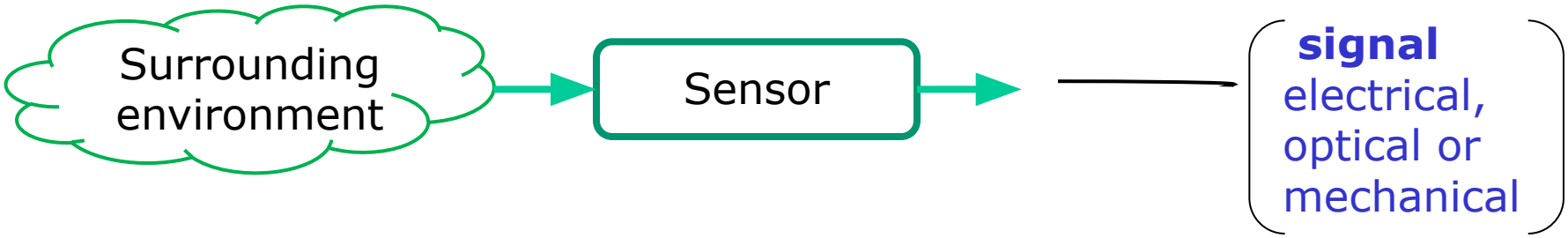
1. Numerical data:
   - Two types based on the values taken:
     - Continuous valued data:
     - Discrete valued data:
2. Categorical data:
   - Three types values they hold:
     - Ordinal values:
     - Nominal values:
     - Binary values:
3. Time series data:

Data Collection

- Data manifests itself in many different forms
- Different forms of data require different ways to collect them and different storage solutions
- Collection of data may consists of sending out surveys, polls or doing other experiments
- Data based on the way it is collected:
  - Data that comes from surveys
    - Usually textual form of data or mixed
  - Data entered in a database as system entry
    - E.g. Student information entered on academic automation system etc.
  - **Data in the form of signals (comes from sensors)**
    - Speech/Audio, Images and videos, Temperature readings, Humidity, Seismic data, EEG (all bio-type signals) etc.
- According to the objective of the task, the way the data is collected will change

Data Collection from Sensors

- Sensors are the devices that respond to the environment around it and convert the physical parameters into a signal (e.g., optical, electrical, mechanical) suitable for processing



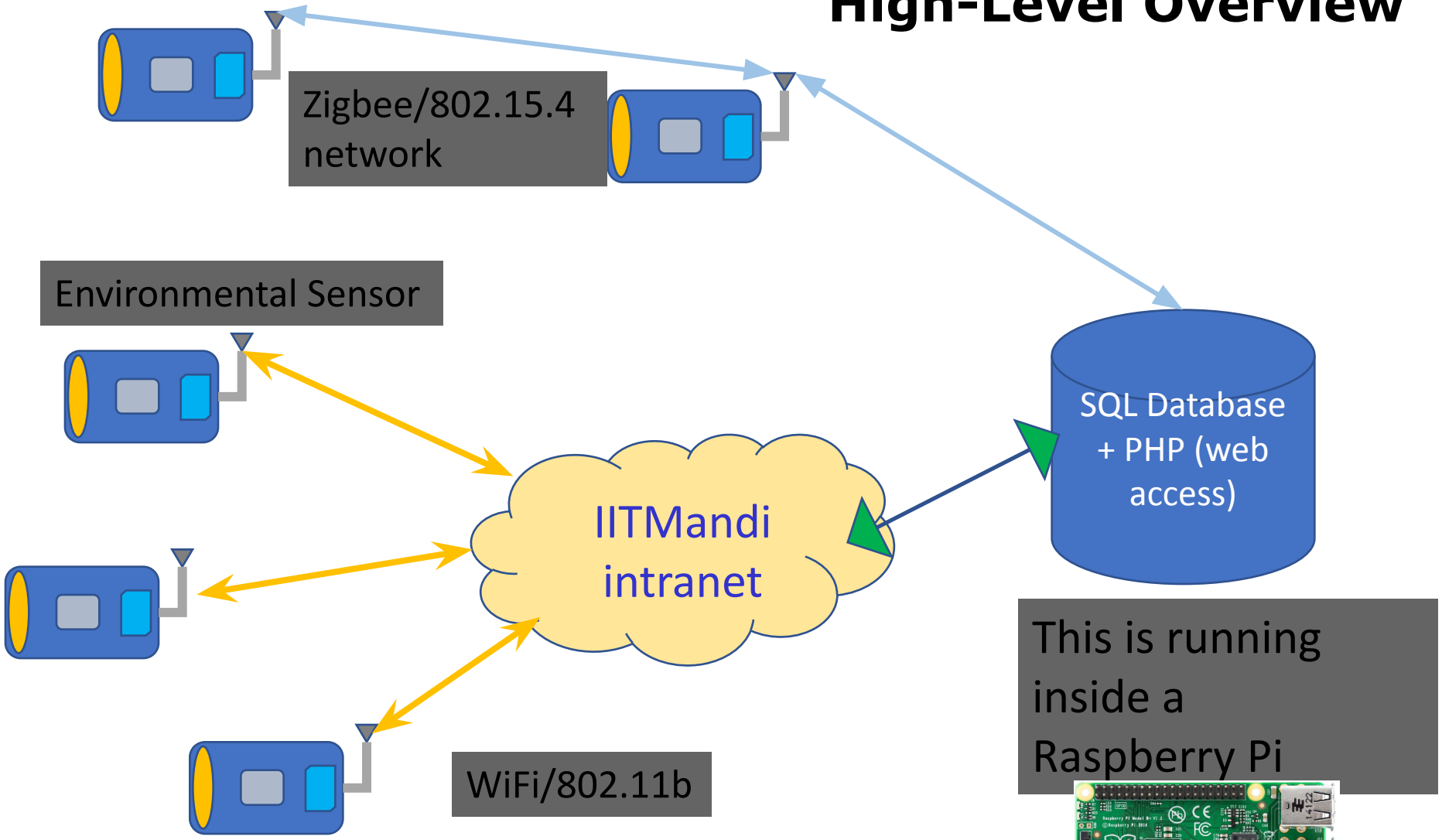
- ***Example***: a temperature sensor outputs an electrical signal whose voltage or current can be used to identify the temperature around it
- Sensors can be an electrical/mechanical component, a module or a subsystem

Different Types of Sensors

- Acoustic, sound sensors (e.g., microphone)
- Visual sensors (e.g. cameras)
- Environmental sensors (e.g., temperature, humidity, pressure etc.)
- Chemical sensors (e.g., Diesel Nitrogen Oxide (Nox) sensors to measure engine-out NOx gas concentration)
- Flow sensors (e.g., water flow sensors)
- Motion sensors (e.g., gyroscope)
- Proximity or presence sensor (e.g., Passive Infrared (PIR))
- Biosensors (e.g., glucose monitor)
- And many more …

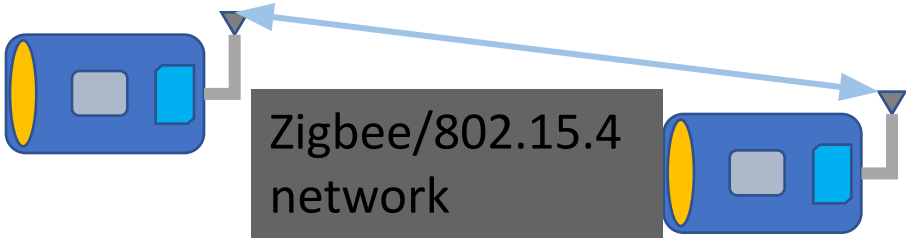
IIT Mandi Weather Station: Environmental Data (Temperature, Humidity, Pressure etc) Collection

High-Level Overview

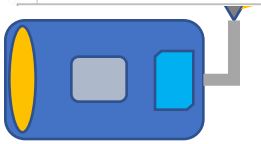


Source: Dr. Siddhartha Sarma

High-Level Overview: Environmental Data (Temperature, Humidity, Pressure etc) Collection

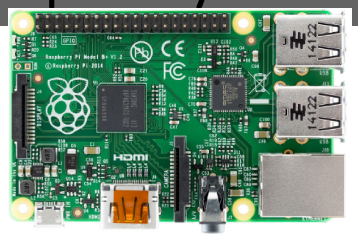


	datestamp	nodeaddr	nodePktId	nodeRSSI	nodeLQI	nodeVolt	tempVal1	tempVal2	tempVal3	humVal	presVal
2											
3	03-11-2017 07:33	fc:c2:3d:00:00:10:ab:fa	1	-53	23	3.027	16.37		16	64	905
4											
5	03-11-2017 07:33	fc:c2:3d:00:00:10:ab:35	2	-84	24	2.905	17.62	17.9794	17	63	904
6											
7	03-11-2017 07:38	fc:c2:3d:00:00:10:ab:fa	3	-54	18	3.027	16.62		16	64	905
8											
9	03-11-2017 07:38	fc:c2:3d:00:00:10:ab:35	4	-84	20	2.905	17.62	17.9794	17	63	904
10											
11	03-11-2017 07:43	fc:c2:3d:00:00:10:ab:fa	5	-50	27	3.027	16.37		16	64	905
12											
13	03-11-2017 07:43	fc:c2:3d:00:00:10:ab:35	6	-86	15	2.905	17.62	18.0789	17	63	904
14											
15	03-11-2017 07:48	fc:c2:3d:00:00:10:ab:fa	7	-52	22	3.027	16.25		16	65	905
16											



WiFi/802.11b

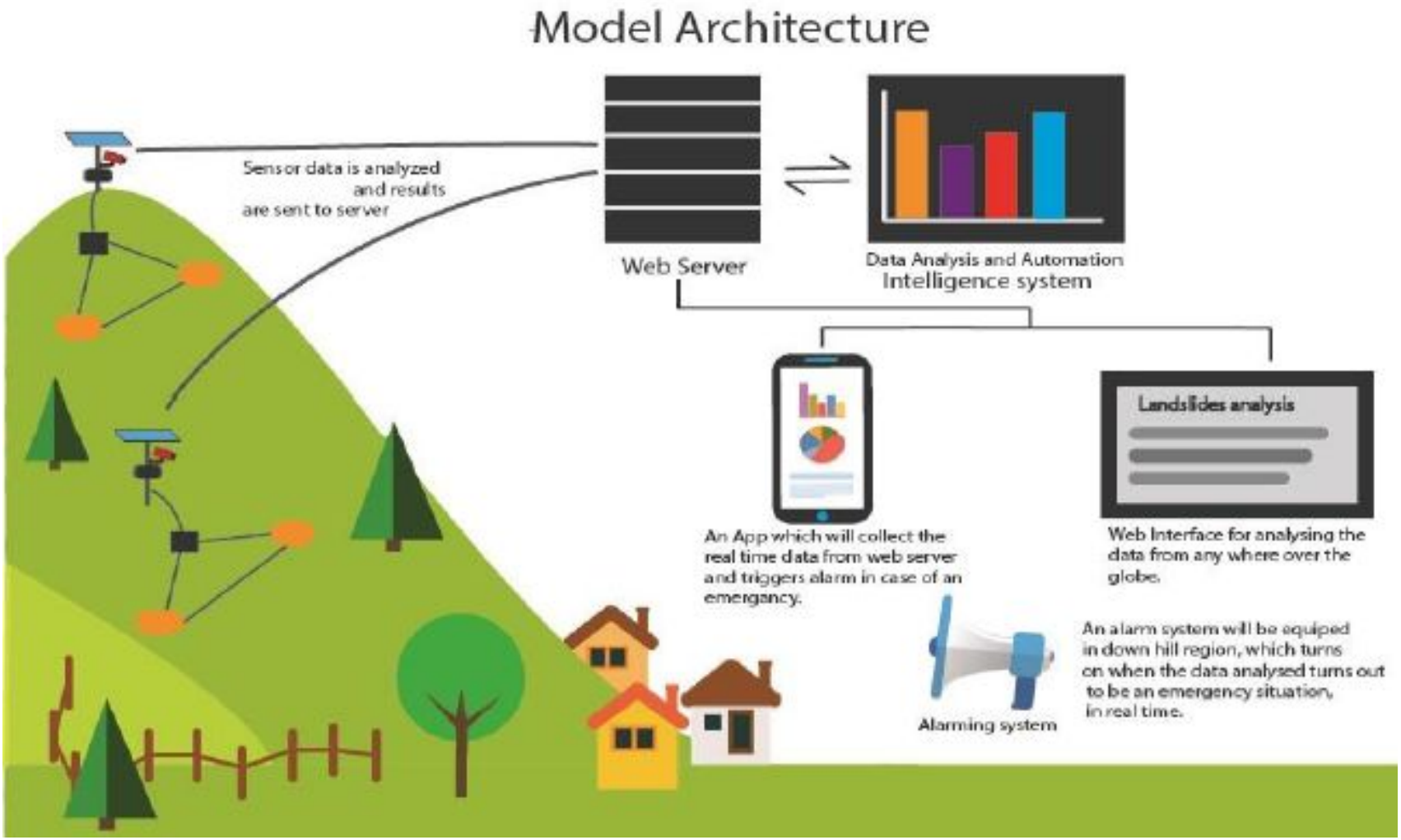
Raspberry Pi



Source: Dr. Siddhartha Sarma

Land Slide Monitoring System (LMS)

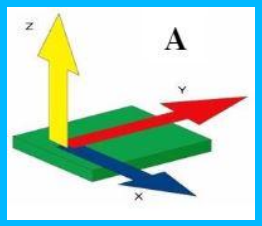
- LMSs that rely on Internet of Things (IoT) and low-cost Micro-Electro-Mechanical Systems (MEMS) sensors



Source: Dr. Varun Dutt (SCEE) and Dr. Uday Kala (SMME)

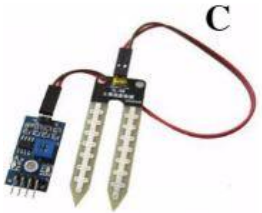
Components of LMS

- The LMS monitors a number of weather and soil parameters via sensors on deployment location



GY 61 Accelerometer Sensor

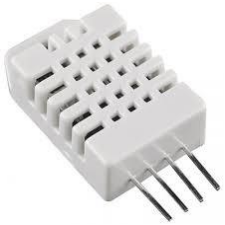
Pin Diagram of GY-61



YL 69 Soil Moisture Sensor

SIM 900A GSM Module

E
Force Sensor



F
Humidity Sensor DHT 22

G
Light Sensor BH-1750

H
Temperature and Pressure Sensor BMP-180

I
Tipping Rain Gauge

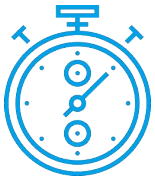
Source: Dr. Varun Dutt (SCEE) and Dr. Uday Kala (SMME)

Architecture and Features of LMS

- The LMS monitors a number of weather and soil parameters via sensors on deployment location

Temperature & Humidity

(-40 C to +80 C & 0-100 %)



Barometric Pressure

(300-1100 mb)

Rainfall Intensity

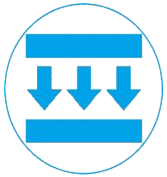
(in mm)

Light Intensity

(0 - 65535 Lux)

Soil movement

(±2000°/sec rotational & ±16g gravitational acceleration)



Soil force

(0-100N)

Soil moisture

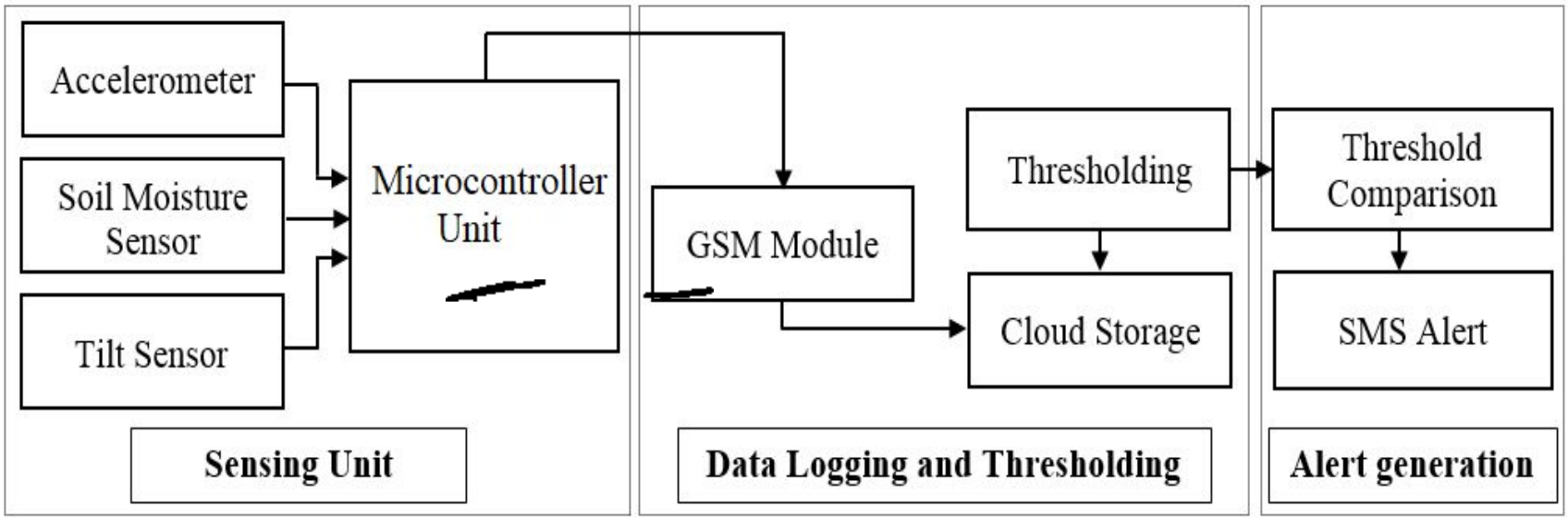
(0-100 %)

Source: Dr. Varun Dutt (SCEE) and Dr. Uday Kala (SMME)

Architecture and Features of LMS

- The LMS monitors a number of weather and soil parameters via sensors on deployment location

Architecture diagram of LMS



The LMS will alert people via traffic lights, SMSs, or smart-apps on mobile phones about the danger of impending landslides

Source: Dr. Varun Dutt (SCEE) and Dr. Uday Kala (SMME)

Architecture and Features of LMS

- The LMS monitors a number of weather and soil parameters via sensors on deployment location

Date/ Time	Temperature (C)/ Humidity (%)	Pressure (Pa)	Rain (inches)	Light intensity (lux)	Accelerations (g)	Force (N)	Moisture (%)
2017-09-06 18:44:32	23.00,56.00	617.64	0.01	3	0.52,0.31,-0.80,0.00,0.00,0.00,31.36,-159.01	0.02	81.00
2017-09-06 18:33:32	24.00,58.00	619.47	0.01	12	0.52,0.30,-0.79,0.00,0.00,0.00,31.45,-159.12	0.02	82.00
2017-09-06 18:22:39	24.00,58.00	623.37	0.00	71	0.52,0.31,-0.80,0.00,0.00,0.00,31.35,-158.88	0.02	83.00
2017-09-06 18:11:31	25.00,60.00	627.02	0.05	194	0.51,0.31,-0.80,0.00,0.00,0.00,30.80,-159.00	0.02	81.00

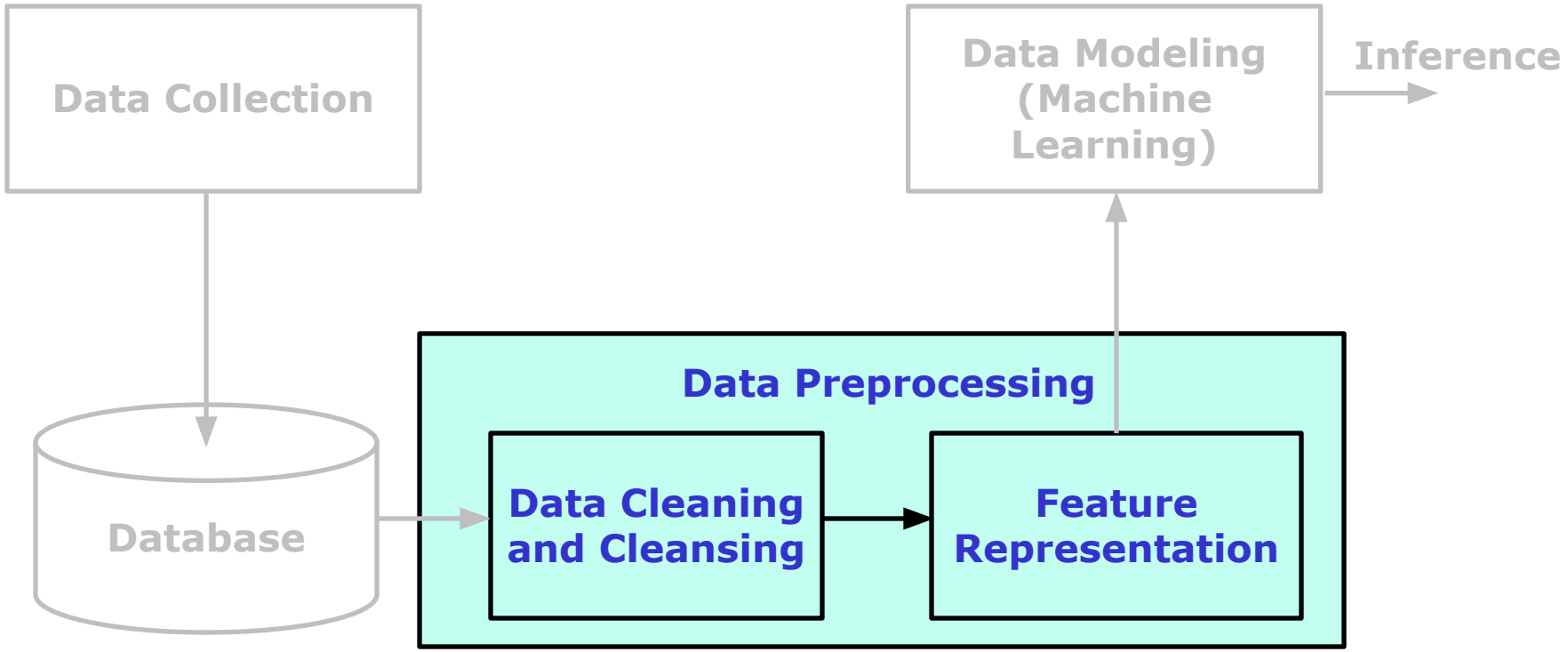
The LMS will alert people via traffic lights, SMSs, or smart-apps on mobile phones about the danger of impending landslides

Source: Dr. Varun Dutt (SCEE) and Dr. Uday Kala (SMME)

Data Preprocessing

Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data
- Machine learning uses data to extract knowledge



Need for Data Preprocessing

- Real world data are tend to be incomplete, noisy and inconsistent due to their huge size and their likely origin from multiple heterogeneous sources
- Preprocessing is important to clean the data
- Low quality data will lead to low quality of analysis results
- If the users believe the data is of low quality (dirty), they are unlikely to trust the results of any data analytics that has been applied to
- Low quality data can cause confusion for analytic procedure using machine learning techniques, resulting in unreliable output
- Incomplete, noisy and inconsistent data are common properties of large real world databases

Tuple (Record) in Structured Data

- A tuple (record) is finite ordered list (sequence) of elements, where each element is belonging to an attribute

Date/ Time	Temperature (C)/ Humidity (%)	Pressure (Pa)	Rain (inches)	Light intensity (lux)	Accelerations (g)	Force (N)	Moisture (%)
2017-09-06 18:44:32	23.00,56.00	617.64	0.01	3	0.52,0.31,-0.80,0.00,0.00,0.00,31.36,-159.01	0.02	81.00
2017-09-06 18:33:32	24.00,58.00	619.47	0.01	12	0.52,0.30,-0.79,0.00,0.00,0.00,31.45,-159.12	0.02	82.00
2017-09-06 18:22:39	24.00,58.00	623.37	0.00	71	0.52,0.31,-0.80,0.00,0.00,0.00,31.35,-158.88	0.02	83.00
2017-09-06 18:11:31	25.00,60.00	627.02	0.05	194	0.51,0.31,-0.80,0.00,0.00,0.00,30.80,-159.00	0.02	81.00

Tuple (record)

- Each row is a tuple

Incomplete Data

- Many tuple (records) have no recorded value for several attributes
- Example:

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	—	—	83.14912	—
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	24.29851	87.68657	963
6	08-07-2018	t11			
7	09-07-2018	t11	26.8494	61.10241	15
8	10-07-2018	t11	27.88806	75.07463	13583.25
9	11-07-2018	t11	27.35915	76.02113	19768.5
10	23-07-2018	t12	24.39024	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.75
12	25-07-2018				
13	26-07-2018	t12	22.19718	99	864

Incomplete Data

- Many tuple (records) have no recorded value for several attributes
- Reasons for incomplete data:
  - User forgot to fill in a field
  - User chose not to fill out the field as it was not considered important at the time of the entry
  - Relevant data may not be recorded due to malfunctioning of equipment
  - Data might have lost while transferring from recorded place
  - Data may not be recorded due to programming error
  - Data might not be recorded due to technology limitations like limited memory

Noisy Data

- Many tuple (records) have incorrect value for several attributes
- Reasons for noisy data:
  - There may be human or computer error occurring in data entry
  - The data collection instruments used may be faulty
  - Error in data transmission
  - There may be technology limitation such as limited buffer size for coordinating synchronised data transfer and consumption

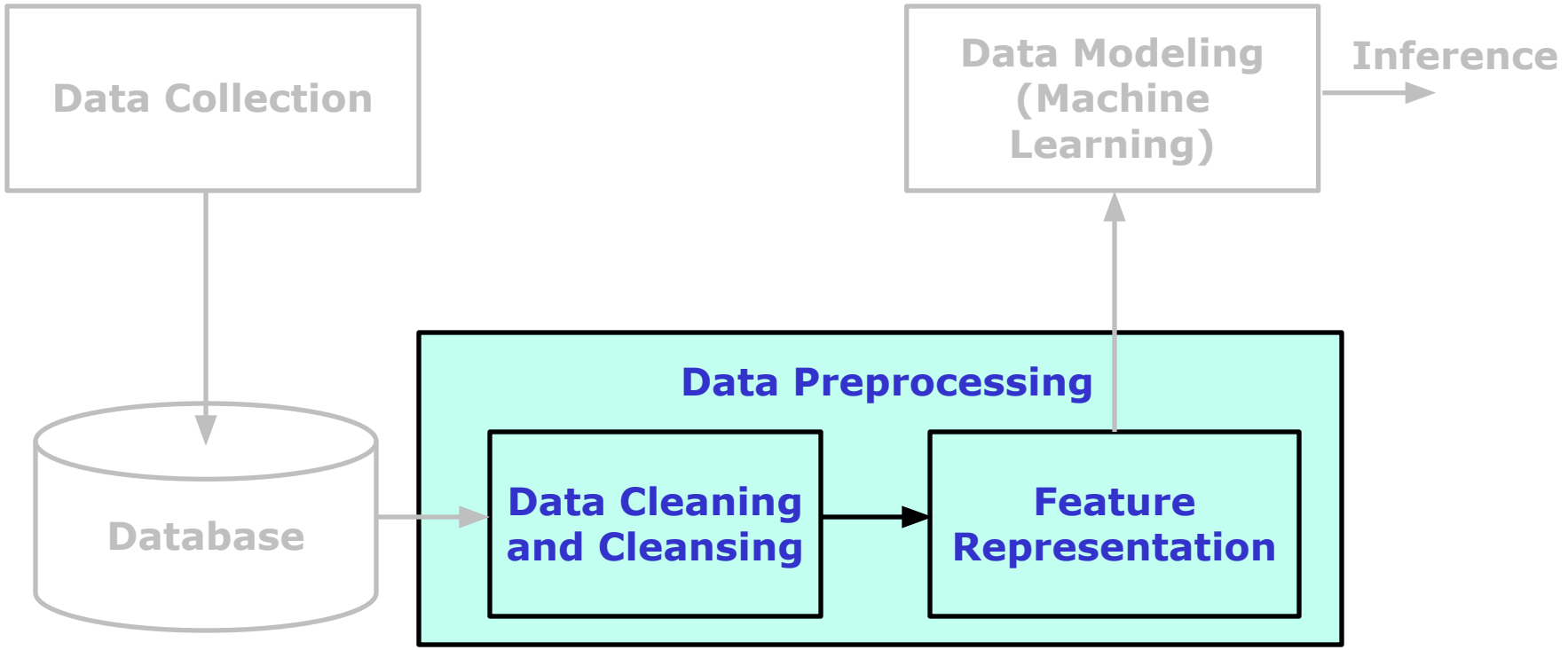
Inconsistent Data

- Data containing discrepancies in stored values for some attributes
- Reasons for inconsistent data:
  - It may result from inconsistencies in
    - name conventions or
      - Example: "Dept_ID", "Department_ID"
        "Roll_No", "Registation_No"
    - data codes used (mismatch in writing values) or
      - Example: For department – "SCEE", "School of Computing and EE"
    - inconsistent formats of input fields such as date
      - Example: "dd-mm-yy", "dd-mm-yyyy", "mm/dd/yyyy"
  - Inconsistency in name convention or formats of input fields while integrating
    - Example: While Integrating temperature records from different locations, if the name conventions are different
  - Inconsistent data may be due to human or computer error occurring in data entry

Data Preprocessing

Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data
- Machine learning uses data to extract knowledge



Need for Data Preprocessing

- Real world data are tend to be incomplete, noisy and inconsistent due to their huge size and their likely origin from multiple heterogeneous sources
- Preprocessing is important to clean the data
- Low quality data will lead to low quality of analysis results
- If the users believe the data is of low quality (dirty), they are unlikely to trust the results of any data analytics that has been applied to
- Low quality data can cause confusion for analytic procedure using machine learning techniques, resulting in unreliable output
- Data could be
  - Incomplete,
  - noisy and
  - inconsistent
  - These are common properties of large real world databases

Data Preprocessing Techniques

- Data cleaning:
- Data integration:
- Data transformation:
- Data reduction :

Data Preprocessing Techniques

- Data cleaning:
  - Applied to
    - identify the missing values,
    - fill in missing values,
    - remove noise and
    - correct inconsistency in the data
- Data integration:
  - It merges data from multiple sources in to a coherent data source
- Data transformation:
  - Transforming the entries of data to a common format
  - Techniques like normalization and standardization applied to transform the data to another form to improve the accuracy and efficiency of machine learning (ML) algorithms involving distance measures

Data Preprocessing Techniques

- Data reduction:
  - Applied to obtain a reduced representation that is much smaller in volume, yet producing almost same analytical results
  - It can reduce the data size by
    - Aggregation
    - Eliminating irrelevant and redundant features (attributes) through correlation analysis
    - Reducing dimension
- *These techniques are not mutually exclusive; they may work together*

Descriptive Data Summarization (Descriptive Analytics)

- It serves as a foundation for data preprocessing
- It helps us to study the general characteristics of data and identify the presence of noise or outliers
- Data characteristics:
  - Central tendency of data
    - Centre of the data
    - Measuring mean, median and mode
  - Dispersion of data
    - The degree to which numerical data tend to spread
    - Measuring range, quartiles, interquartile range (IQR), the five-number summary and standard deviation

Descriptive Analytics: Measuring Central Tendency

- Mean:
  - Let x_1, x_2, …, x_N be a set of N values in an attribute. Mean of this set of values is given by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Number of records (tuples), N = 10

Years of experience	Salary (in Rs 1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83
Sum:	**91**

Descriptive Analytics: Measuring Central Tendency

- Mean:
  - Let x_1, x_2, …, x_N be a set of N values in an attribute. Mean of this set of values is given by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Number of records (tuples), N = 10

Years of experience	Salary (in Rs 1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Mean Years of experience: Sum/10 **9.1**

Descriptive Analytics: Measuring Central Tendency

- Mean:
  - Let x_1, x_2, …, x_N be a set of N values in an attribute. Mean of this set of values is given by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Number of records (tuples), N = 10

Years of experience	Salary (in Rs 1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Mean Salary: Sum/10

55.4

Descriptive Analytics: Measuring Central Tendency

- Mean:
  - Let x_1, x_2, …, x_N be a set of N values in an attribute. Mean of this set of values is given by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

  - Mean is a better measure of central tendency for the symmetric data (symmetrically distributed data)

Number of records (tuples), N = 10

Years of experience	Salary (in Rs 1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83
Mean:	**9.1**	**55.4**

Descriptive Analytics: Measuring Central Tendency

- Median:
  - Let x_1, x_2, …, x_N be a set of N values in an attribute. The median is the "middle" number (value), when those numbers are listed in order from smallest to greatest.
  - Median is the value separating the higher half from the lower half of a data sample
  - For a given data of N values in sorted order
    - If N is odd, then median is the middle value of the ordered list
    - If N is even, then median is the average of middle two values

Number of records (tuples), N = 10

Years of experience	Salary (in Rs 1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Illustration: Median of attribute "Years of experience"

Descriptive Analytics: Measuring Central Tendency

- Median:
  - Let x_1, x_2, …, x_N be a set of N values in an attribute. The median is the "middle" number (value), when those numbers are listed in order from smallest to greatest.
  - Median is the value separating the higher half from the lower half of a data sample
  - For a given data of N values in sorted order
    - If N is odd, then median is the middle value of the ordered list
    - If N is even, then median is the average of middle two values

Sort the values in "Years of experience"

Years of experience
1
3
6
8
9
11
13
16
16
21

Descriptive Analytics: Measuring Central Tendency

- Median:
  - Let x_1, x_2, …, x_N be a set of N values in an attribute. The median is the "middle" number (value), when those numbers are listed in order from smallest to greatest.
  - Median is the value separating the higher half from the lower half of a data sample
  - For a given data of N values in sorted order
    - If N is odd, then median is the middle value of the ordered list
    - If N is even, then median is the average of middle two values

Sort the values in "Years of experience"

Years of experience
1
3
6
8
9
11
13
16
16
21

Median: $\frac{9+11}{2}$

Descriptive Analytics: Measuring Central Tendency

- Median:
  - Let x_1, x_2, …, x_N be a set of N values in an attribute. The median is the "middle" number (value), when those numbers are listed in order from smallest to greatest.
  - Median is the value separating the higher half from the lower half of a data sample
  - For a given data of N values in sorted order
    - If N is odd, then median is the middle value of the ordered list
    - If N is even, then median is the average of middle two values
  - For asymmetrically distributed (skewed) data, a better measure of centre of data is median

Sort the values in "Years of experience"

Years of experience
1
3
6
8
9
11
13
16
16
21

Median: **10**

Descriptive Analytics: Measuring Central Tendency

- Mode: Most frequent value in an attribute in the data

Illustration: Mode of attribute "Years of experience"

Assume that values are discrete numerical

Number of records (tuples), N = 10

Years of experience	Salary (in Rs 1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Mode: **3**

Descriptive Analytics: Measuring Central Tendency

- Mode: Most frequent value in an attribute in the data

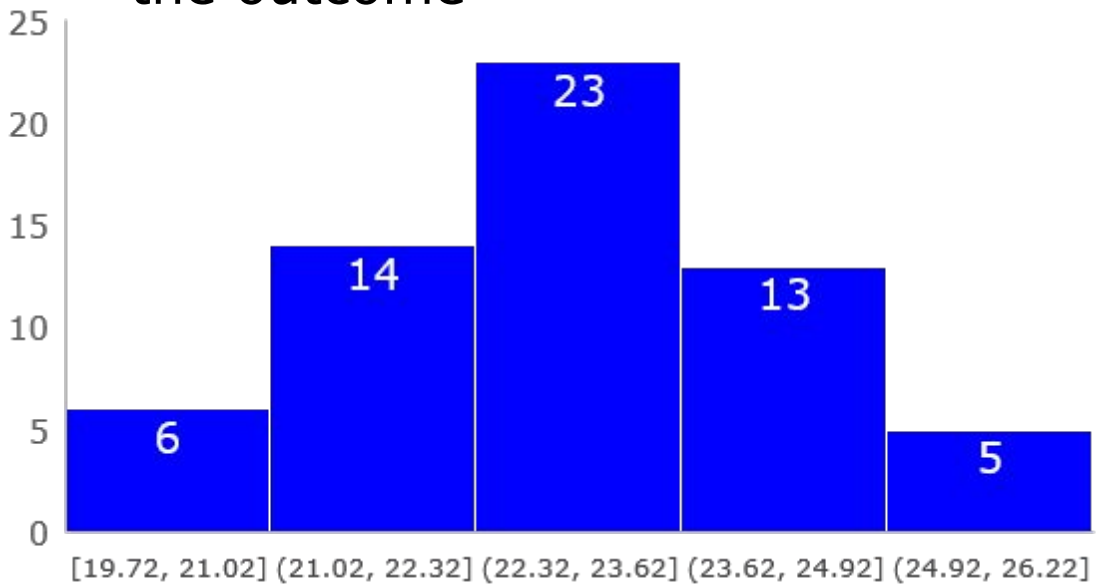
Number of samples, N = 61

Date	Temperature
Sept 1	25.47
Sept 2	26.19
Sept 3	25.17
Sept 4	24.30
Sept 5	24.07
Sept 6	21.21
Sept 7	23.49
Sept 8	21.79
Sept 9	25.09
Sept 10	25.39
---	---
Oct 29	23.06
Oct 30	23.72
Oct 31	23.02

Mean: **22.85**

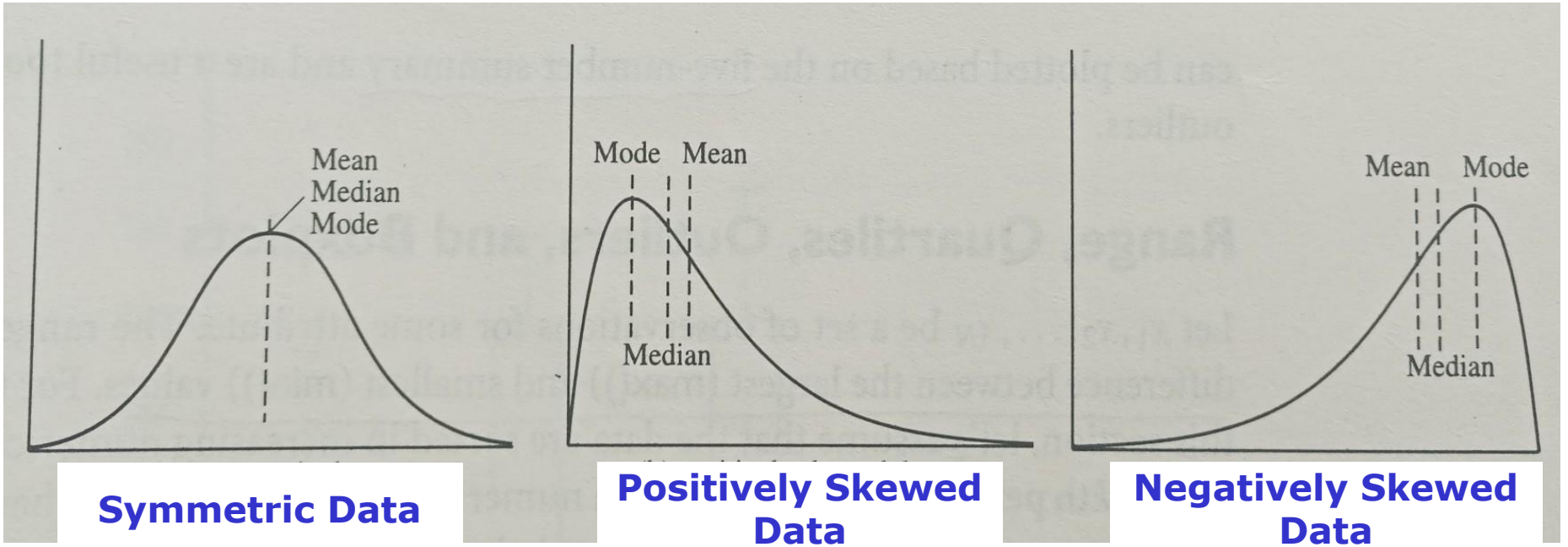
Median: **22.89**

- The mode of a continuous variable is the value at which the probability density function, f(x) , is at a maximum.
- It is a value that is most likely to lie within the same interval as the outcome



Mode: (22.32 – 23.62] ~=22.97

Descriptive Analytics: Measuring Central Tendency



Symmetric Data

Positively Skewed Data

Negatively Skewed Data

Descriptive Analytics: Measuring Dispersion of Data

- The degree to which numerical data tend to spread
- It is also called as variance (in symmetrically distributed data)
- Common measures of data dispersion:
  - Range
  - The five-number summery (based on quartiles)
  - The inter quartile range (IQR)
  - Standard deviation
- Range: The range of a finite set of values is the difference between the maximum and minimum values

Descriptive Analytics: Measuring Dispersion of Data

- Quartiles:
  - The k^{th} percentile:
    - Let x_1, x_2, …, x_N be a set of N values in an attribute
    - The k^{th} percentile of a set of data in numerical order is the value of x_n having the property that k percent of data entries lie at or below x_n
  - Example: 50^{th} percentile
    - The value (number) below which 50% of the data entries (values) lie
      - Those 50% of entries have values equal to or less that 50^{th} percentile

Number of records (tuples), N = 10

Years of experience	Salary (in Rs 1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Illustration: 50th percentile of attribute "Years of experience"

Descriptive Analytics: Measuring Dispersion of Data

- Quartiles:
  - The k^{th} percentile:
    - Let $x_1, x_2, \ldots, x_N$ be a set of N values in an attribute
    - The k^{th} percentile of a set of data in numerical order is the value of x_n having the property that k percent of data entries lie at or below x_n
  - Example: 50^{th} percentile
    - The value (number) below which 50% of the data entries (values) lie
      - Those 50% of entries have values equal to or less that 50^{th} percentile

Sort the values in "Years of experience"

Years of experience
1
3
6
8
9
11
13
16
16
21

50^{th} Percentile: 10

Illustration: 50^{th} percentile of attribute "Years of experience"

Descriptive Analytics: Measuring Dispersion of Data

- Quartiles:
  - The k^{th} percentile:
    - Let x_1, x_2, …, x_N be a set of N values in an attribute
    - The k^{th} percentile of a set of data in numerical order is the value of x_n having the property that k percent of data entries lie at or below x_n
  - Example: 25th percentile
    - The value (number) below which 25% of the data entries (values) lie
      - Those 25% of entries have values equal to or less that 25th percentile
    - Middle element between minimum and 50th percentile

Sort the values in "Years of experience"

Years of experience
1
3
6
8
9
11
13
16
16
21

25th Percentile: **6**

Illustration: 25th percentile of attribute "Years of experience"

Descriptive Analytics: Measuring Dispersion of Data

- Quartiles:
  - The k^{th} percentile:
    - Let x_1, x_2, …, x_N be a set of N values in an attribute
    - The k^{th} percentile of a set of data in numerical order is the value of x_n having the property that k percent of data entries lie at or below x_n
  - Example: 75^{th} percentile
    - The value (number) below which 75% of the data entries (values) lie
      - Those 75% of entries have values equal to or less that 75^{th} percentile
    - Middle element between maximum and 50^{th} percentile

Sort the values in "Years of experience"

Years of experience
1
3
6
8
9
11
13
16
16
21

75^{th} Percentile: 16

Illustration: 75^{th} percentile of attribute "Years of experience"

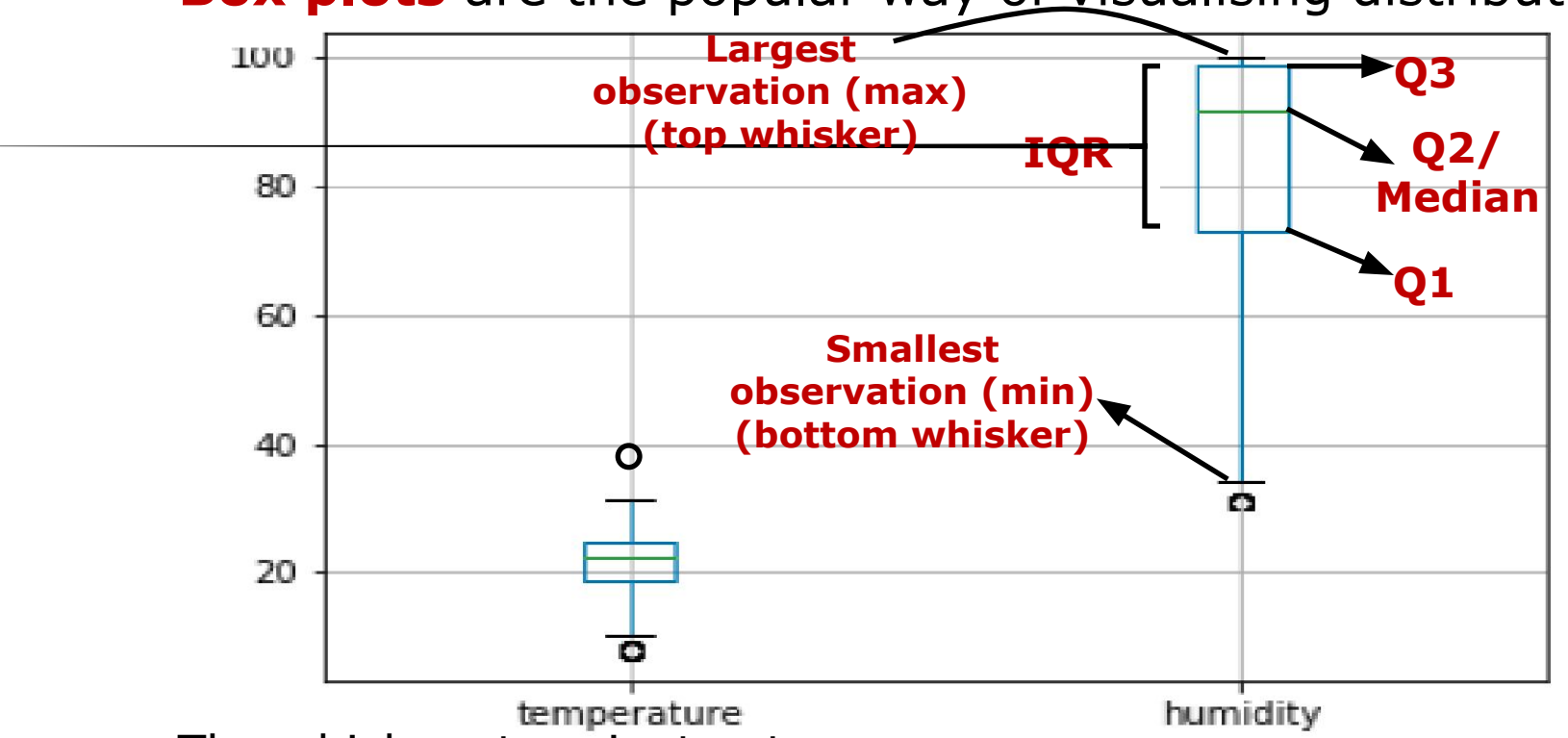
Descriptive Analytics: Measuring Dispersion of Data

- Quartiles:
  - The k^{th} percentile:
    - Let $x_1, x_2, \ldots, x_N$ be a set of N values in an attribute
    - The k^{th} percentile of a set of data in numerical order is the value of x_n having the property that k percent of data entries lie at or below x_n
    - Median is the 50^{th} percentile (the second quartile (Q2))
    - The first quartile (Q1): It is the 25^{th} percentile
    - The third quartile (Q3): It is the 75^{th} percentile
  - The quartiles including median give some indication of centre, spread and shape of distribution
- The distance between the Q1 and Q3 is a simple measure of spread
- Interquartile range (IQR): Distance between the first quartile (Q1) and third quartile (Q2)

$$IQR = Q3 - Q1$$

Descriptive Analytics: Measuring Dispersion of Data

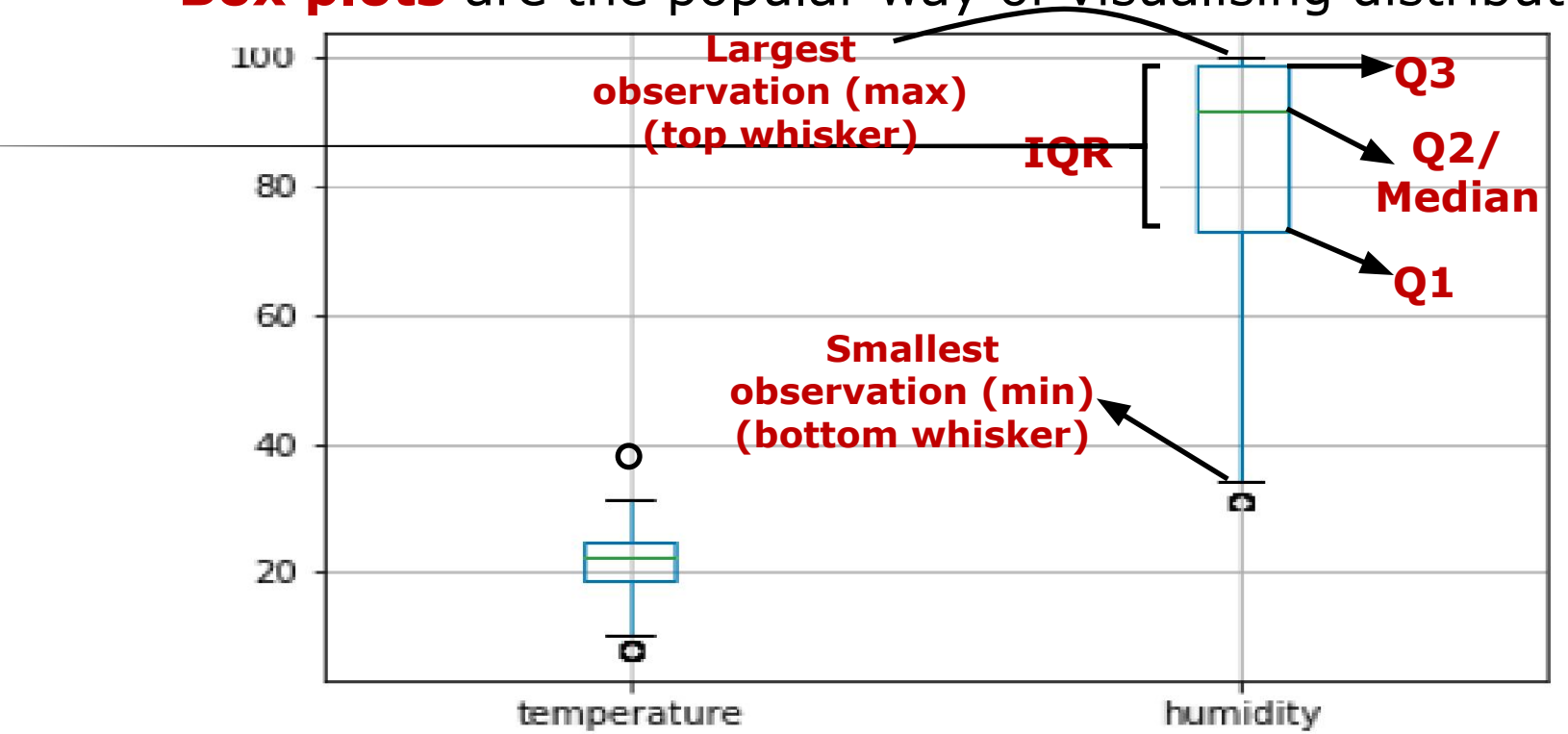
- The five-number summary of distribution:
  - It consists of minimum value, Q1, median, Q3 and maximum value
- **Box plots** are the popular way of visualising distribution



- The whiskers terminate at
  - Smallest (minimum) or largest (maximum) observations **or**
  - the most extreme observations occurring within 1.5 x IQR of respective quartiles (Q1 and Q3)

Descriptive Analytics: Measuring Dispersion of Data

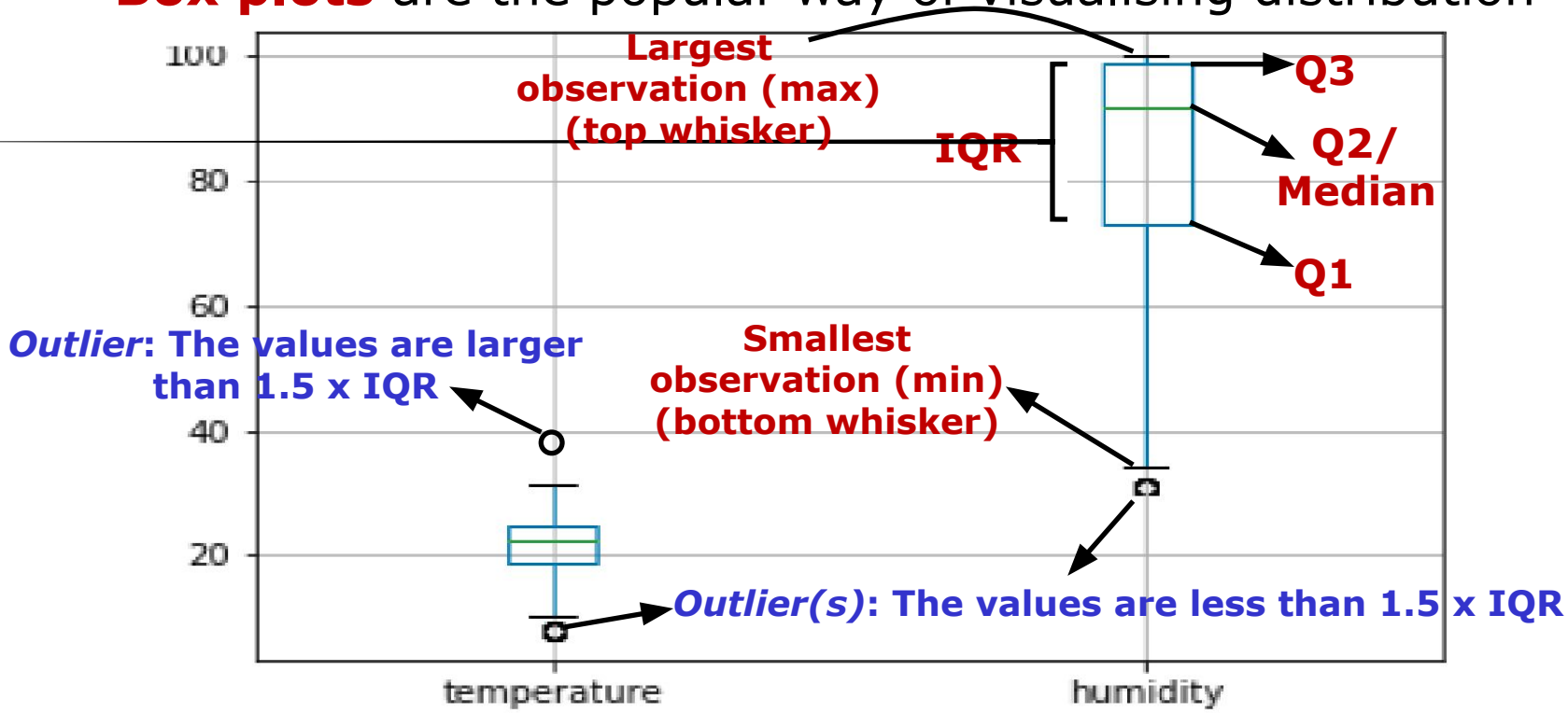
- The five-number summary of distribution:
  - It consists of minimum value, Q1, median, Q3 and maximum value
- **Box plots** are the popular way of visualising distribution



- 1.5 x IQR is equivalent to 2.7σ from mean if the distribution is normal distribution
  - It is close to 3σ from mean which is a standard in normal distribution

Descriptive Analytics: Measuring Dispersion of Data

- The five-number summary of distribution:
  - It consists of minimum value, Q1, median, Q3 and maximum value
- **Box plots** are the popular way of visualising distribution



- **Lower bound: Q1 – (1.5 x IQR)** **Upper bound: Q3 + (1.5 x IQR)**
- Outliers: Any datapoint less than the lower bound and larger than the upper bound

Descriptive Analytics: Measuring Dispersion of Data

- Variance (σ^2):
  - Let x_1, x_2, …, x_N be a set of N values in an attribute. variance (σ^2) of this set of values is given by

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2 \qquad \mu = \text{mean}$$

- Standard deviation (σ):
  - The square root of variance $\sigma = \sqrt{\text{Variance}}$
- Standard deviation measures the spread about the mean
  - It is used when the mean is chosen as the measure of centre, especially in symmetric distribution
- The quartiles Q1 and Q3 measure the spread about median
  - Q1 and Q3 are used when the median is chosen as the measure of centre, especially in skewed distribution

Data Preprocessing

Data Cleaning: Handling Missing Values, Noisy Data and Outliers

Data Cleaning (Data Cleansing)

- Real world data are tend to be incomplete, noisy and inconsistent
- Data cleaning routines attempt to identify missing values, fill in missing values, smooth out noise while identifying outliers and correct inconsistencies in the data

> *80 percent of a data scientist's valuable time is spent simply finding, cleansing, and organizing data, leaving only 20 percent to actually perform analysis...*
>
> IBM Data Analytics

- One of the biggest data cleaning task is handling missing values

Data Cleaning: Missing Values

- Many tuple (records) have no recorded value for several attributes
- Identifying missing values:
  - When Pandas library for python is used, it detect the missing values as "NaN" [1]
  - It automatically consider "blank" in the attribute value, "NaN/nan/NAN" in the attribute value , "NA" in the attribute value, "n/a" in the attribute value, "NULL/null" in the attribute value as NaN

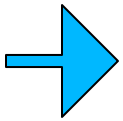
- Important note: If any numeric attribute have the value 0 (zero), then it is *not a missing value*
  - If it is not correct value, then it is *simply a noise*

[1] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html

Methods to Handle Missing Values

- Ignore the tuples:
  - This method is effective only when the tuples contain several attributes (> 50% of attributes) with missing value

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018			83.14912	
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	24.29851	87.68657	963
6	08-07-2018	t11			
7	09-07-2018	t11	26.8494	61.10241	15
8	10-07-2018	t11	27.88806	75.07463	13583.25
9	11-07-2018	t11	27.35915	76.02113	19768.5
10	23-07-2018	t12	24.39024	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.75
12	25-07-2018				
13	26-07-2018	t12	22.19718	99	864



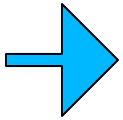
	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	10-07-2018	t10	25.17021	85.34043	652.5
4	11-07-2018	t10	24.29851	87.68657	963
5	09-07-2018	t11	26.8494	61.10241	15
6	10-07-2018	t11	27.88806	75.07463	13583.25
7	11-07-2018	t11	27.35915	76.02113	19768.5
8	23-07-2018	t12	24.39024	94.4065	1071
9	24-07-2018	t12	24.16197	97.66901	438.75
10	26-07-2018	t12	22.19718	99	864

Tuples contain several attributes (> 50% of attributes) with missing value

Methods to Handle Missing Values

- Ignore the tuples:
  - This method is effective only when the tuples contain several attributes (> 50% of attributes) with missing value
  - This method is also used when the target variable (class label) is missing

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	83.14912	
4	10-07-2018	NaN	25.17021	85.34043	652.5
5	11-07-2018	t10	24.29851	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494		15
8	10-07-2018	t11	27.88806	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12		94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	NaN	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864



	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	83.14912	
4	11-07-2018	t10	24.29851	87.68657	963
5	08-07-2018	t11	23.53846	61.92308	3
6	09-07-2018	t11	26.8494		15
7	10-07-2018	t11	27.88806	75.07463	13583
8	11-07-2018	t11	27.35915	76.02113	19769
9	23-07-2018	t12		94.4065	1071
10	24-07-2018	t12	24.16197	97.66901	438.8
11	26-07-2018	t12	22.19718	99	864

Target attribute (StationID) with missing value

Methods to Handle Missing Values

- Fill in the missing values (imputing values) manually:
  - Time consuming
  - Not feasible given a large data set with many missing values

- Use a global constant to fill in missing value (Imputing global constant):
  - Replace all missing attribute values by a same constant
  - Imputed value may not be correct

Methods to Handle Missing Values

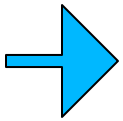
- Use attribute mean/median/mode to fill in the missing value (mean/median/mode imputation):
  - Applicable to numeric data
  - Centre of the data won't change

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	NaN	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	NaN	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	NaN	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

Methods to Handle Missing Values

- Use attribute mean/median/mode to fill in the missing value (mean/median/mode imputation):
  - Applicable to numeric data
  - Centre of the data won't change

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	NaN	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	NaN	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	NaN	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

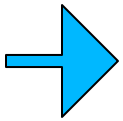


	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.1368	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	25.1368	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	25.1368	94.4065	1071
11	24-07-2018	t12	24.16197	(Ctrl) 01	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

Methods to Handle Missing Values

- Use attribute mean/median/mode to fill in the missing value (mean/median/mode imputation):
  - Applicable to numeric data
  - Centre of the data won't change
  - However, it does not preserve the relationship with other variables

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.1368	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	25.1368	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	25.1368	94.4065	1071
11	24-07-2018	t12	24.16197	(Ctrl) 01	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

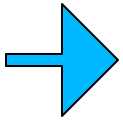


	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	85.42	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.1368	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	85.42	15
8	10-07-2018	t11	25.1368	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	25.1368	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

Methods to Handle Missing Values

- Filling with local mean/median/mode:
  - Use attribute mean/median/mode of all samples belonging to a group (class) to fill in the missing value
    - Applicable to numeric data
    - Centre of the data of a group won't change

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	NaN	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	NaN	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	NaN	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

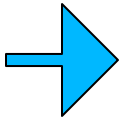


	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.612	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	NaN	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	NaN	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

Methods to Handle Missing Values

- Filling with local mean/median/mode:
  - Use attribute mean/median/mode of all samples belonging to a group (class) to fill in the missing value
    - Applicable to numeric data
    - Centre of the data of a group won't change

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.612	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	NaN	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	NaN	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

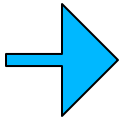


	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.612	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	25.916	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	NaN	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

Methods to Handle Missing Values

- Filling with local mean/median/mode:
  - Use attribute mean/median/mode of all samples belonging to a group (class) to fill in the missing value
    - Applicable to numeric data
    - Centre of the data of a group won't change
    - However, it does not preserve the relationship with other variables

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.612	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	25.916	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	NaN	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

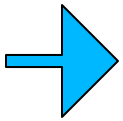


	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.612	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	25.916	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	23.884	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

Methods to Handle Missing Values

- Use the values from the previous/next record (with in a group) to fill in missing value (Padding)
  - Useful only when the domine understanding is good

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	NaN	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	NaN	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	NaN	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864



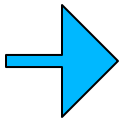
	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	82.1875	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.17021	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	61.92308	15
8	10-07-2018	t11	26.8494	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	24.16197	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

- If the data is categorical or text, one can replace the missing values by most frequent observations

Methods to Handle Missing Values

- Use most probable value to fill the missing value:
  - Use interpolation technique to predict the missing value
    - **Linear interpolation** is achieved by geometrically rendering a straight line between two adjacent points on a graph or plane
    - Interpolation happens column wise
    - Popular strategy
    - It does not preserves the relationship with other variables

	Dates	Temperature	Humidity	Rain
2	08-07-2018	25.46875	82.1875	6.75
3	09-07-2018	26.19298	83.1491	1761.75
4	10-07-2018	25.17021	85.3404	652.5
5	11-07-2018	NaN	87.6866	963
6	12-07-2018	24.06923	87.6462	254.25
7	13-07-2018	21.20779	95.9481	339.75
8	15-07-2018	23.48571	96.1714	38.25
9	18-07-2018	NaN	98.5897	29.25
10	19-07-2018	25.09346	88.3271	4.5
11	20-07-2018	25.39423	90.4327	112.5
12	21-07-2018	NaN	94.5378	735.75
13	22-07-2018	22.5098	99	607.5
14	23-07-2018	22.904	98	717.75
15	24-07-2018	NaN	99	513
16	25-07-2018	23.18182	98.9697	195.75

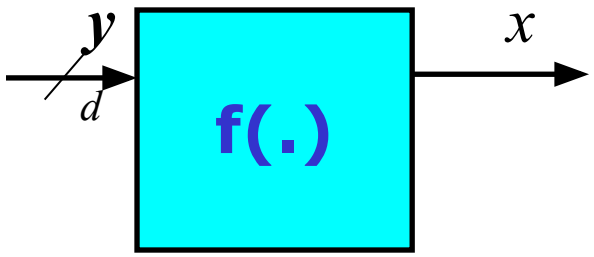


	Dates	Temperature	Humidity	Rain
2	08-07-2018	25.46875	82.1875	6.75
3	09-07-2018	26.19298	83.1491	1761.75
4	10-07-2018	25.17021	85.3404	652.5
5	11-07-2018	24.2	87.6866	963
6	12-07-2018	24.06923	87.6462	254.25
7	13-07-2018	21.20779	95.9481	339.75
8	15-07-2018	23.48571	96.1714	38.25
9	18-07-2018	21.5	98.5897	29.25
10	19-07-2018	25.09346	88.3271	4.5
11	20-07-2018	25.39423	90.4327	112.5
12	21-07-2018	23.7	94.5378	735.75
13	22-07-2018	22.5098	99	607.5
14	23-07-2018	22.904	98	717.75
15	24-07-2018	21.6	99	513
16	25-07-2018	23.18182	98.9697	195.75

Methods to Handle Missing Values

- Use most probable value to fill the missing value:
  - Use regression techniques to predict the missing value (regression imputation)
    - Let y_1, y_2, …, y_d be a set of d attributes
    - Regression (multivariate): The n^{th} value is predicted as

$$x_n = f(y_{n1}, y_{n2}, \ldots, y_{nd})$$



    - Linear regression (multivariate): $x_n = w_1 y_{n1} + w_2 y_{n2} + \ldots + w_d y_{nd}$

Methods to Handle Missing Values

- Use most probable value to fill the missing value:
  - Use regression techniques to predict the missing value (regression imputation)
    - Let y_1, y_2, …, y_d be a set of d attributes
    - Regression (multivariate): The n^{th} value is predicted as

$$x_n = f(y_{n1}, y_{n2}, \ldots, y_{nd})$$

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	NaN	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	NaN	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	NaN	15
8	10-07-2018	t11	NaN	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	NaN	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

$$Temperature = f(Humidity, Rain)$$

$$Temperature = w_{T1} Humidity + w_{T2} Rain$$

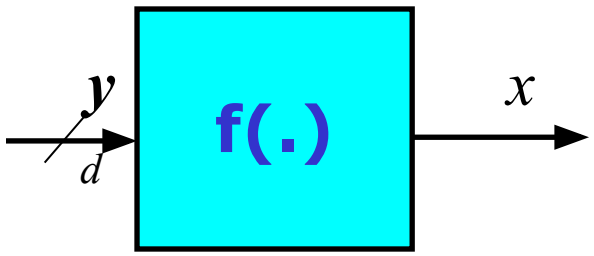
$$Humidity = f(Temperature, Rain)$$

$$Humidity = w_{H1} Temperature + w_{H2} Rain$$

Methods to Handle Missing Values

- Use most probable value to fill the missing value:
  - Use regression techniques to predict the missing value (regression imputation)
    - Let y_1, y_2, …, y_d be a set of d attributes
    - Regression (multivariate): The n^{th} value is predicted as

$$x_n = f(y_{n1}, y_{n2}, \ldots, y_{nd})$$



    - Linear regression (multivariate): $x_n = w_1 y_{n1} + w_2 y_{n2} + \ldots + w_d y_{nd}$
    - Popular strategy
    - It uses the most information from the present data to predict the missing values
    - *It preserves the relationship with other variables*

Data Cleaning: Handling the Noisy Data

- Noise is a random error or variance in a measured variable
- Consider the case where most of the entries in a numeric attribute is 0 (zero)
- Example1

Date	Temperature
Sept 1	25.47
Sept 2	26.19
Sept 3	0
Sept 4	24.30
Sept 5	24.07
Sept 6	21.21
Sept 7	0
Sept 8	21.79
Sept 9	25.09
Sept 10	0
---	---

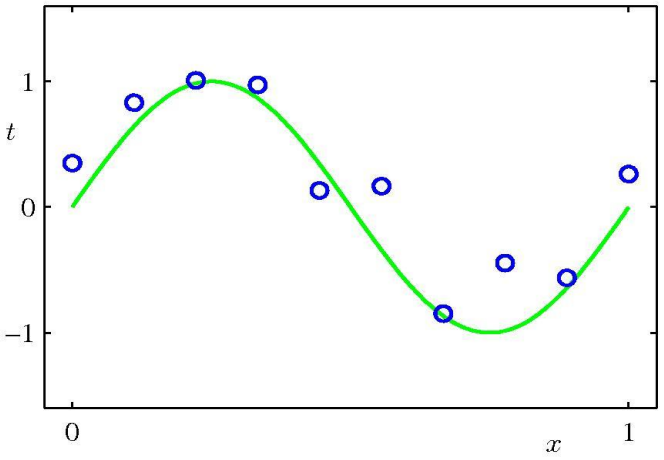
- Example2: Pima-Indians-Diabetes

---	BMI	Age	---
---	33.6	50	---
---	26.6	31	---
---	23.3	32	---
---	0	21	---
---	43.1	33	---
---	25.6	30	---
---	0	26	---
---	35.3	29	---
---	30.5	53	---
---	0	54	---

- Replace the 0s (zeros) based on domain knowledge
- Replace the 0s (zeros) by regression based methods

Data Cleaning: Smoothing the Noisy Data

- Noise is a random error or variance in a measured variable
- Due to noise, many tuple (records) have incorrect value for several attributes
- Mostly data is full of noise
- Smooth out the data to remove the effect of noise
- Data smoothing allows important patterns to stand out
- The idea is to sharpen the patterns (values) in the data and highlight trends the data is pointing to



- Methods for data smoothing:
  - Binning
  - Regression (function approximation)

Binning Methods for Data Smoothing

- Binning method smooth a sorted data value of a noisy attribute by consulting its neighbourhood i.e., the values around it
- It perform local smoothing as this method consult the neighbourhood of values
- The sorted values are partitioned into (almost) equal-frequency bins

Binning Methods for Data Smoothing

- ***Different approaches for smoothing by bin:***

1. Smoothing by bin means:
   - Each value in a bin is replaced by the mean value of the bin
2. Smoothing by bin medians:
   - Each value in a bin is replaced by the median value of the bin
3. Smoothing by bin boundaries:
   - The minimum and maximum values in a given bin are identified as bin boundaries
   - Each bin value is then replaced by the closest boundary value

- Larger the width, the greater the effect of the smoothing

Illustration of Binning Methods for Data Smoothing

- Example:
- Noisy data for price (in Rs) : 8, 15, 34, 24, 4, 21, 28, 21, 25
- Sorted data for price (in Rs) : 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into bins:

Bin1: 4, 8, 15

Bin2: 21, 21, 24

Bin3: 25, 28, 34

Smoothing by bin means:

Bin1: 9, 9, 9

Bin2: 22, 22, 22

Bin3: 29, 29, 29

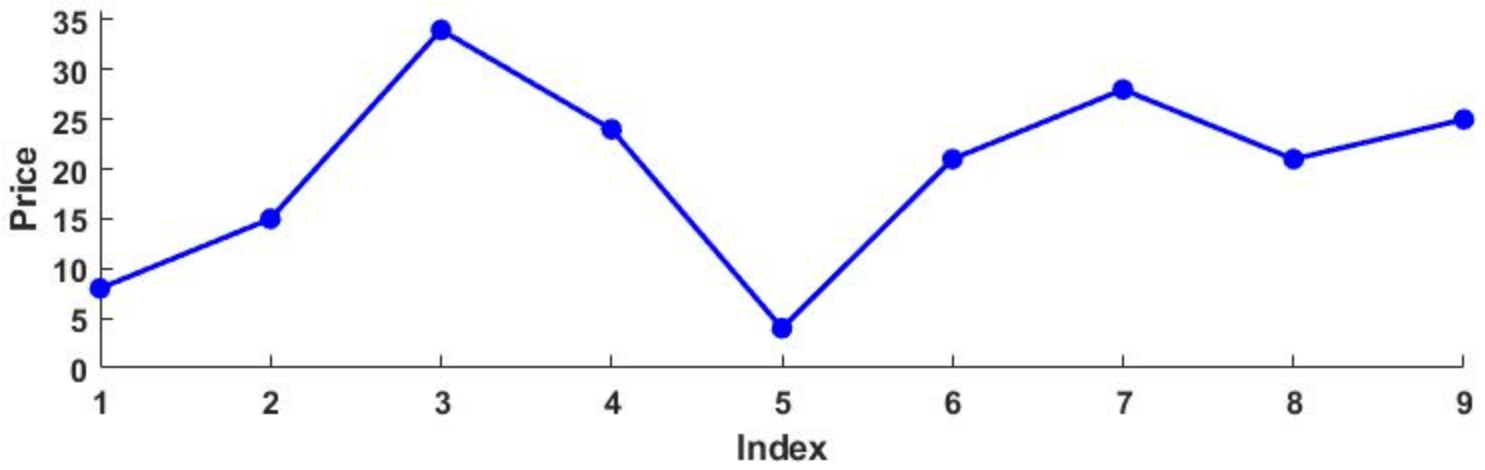


Illustration of Binning Methods for Data Smoothing

- Example:
- Noisy data for price (in Rs) : 8, 15, 34, 24, 4, 21, 28, 21, 25
- **Smoothing by bin means** : 9, 9, 29, 22, 9, 22, 29, 22, 29

Partition into bins:

Bin1: 4, 8, 15

Bin2: 21, 21, 24

Bin3: 25, 28, 34

Smoothing by bin means:

Bin1: 9, 9, 9

Bin2: 22, 22, 22

Bin3: 29, 29, 29

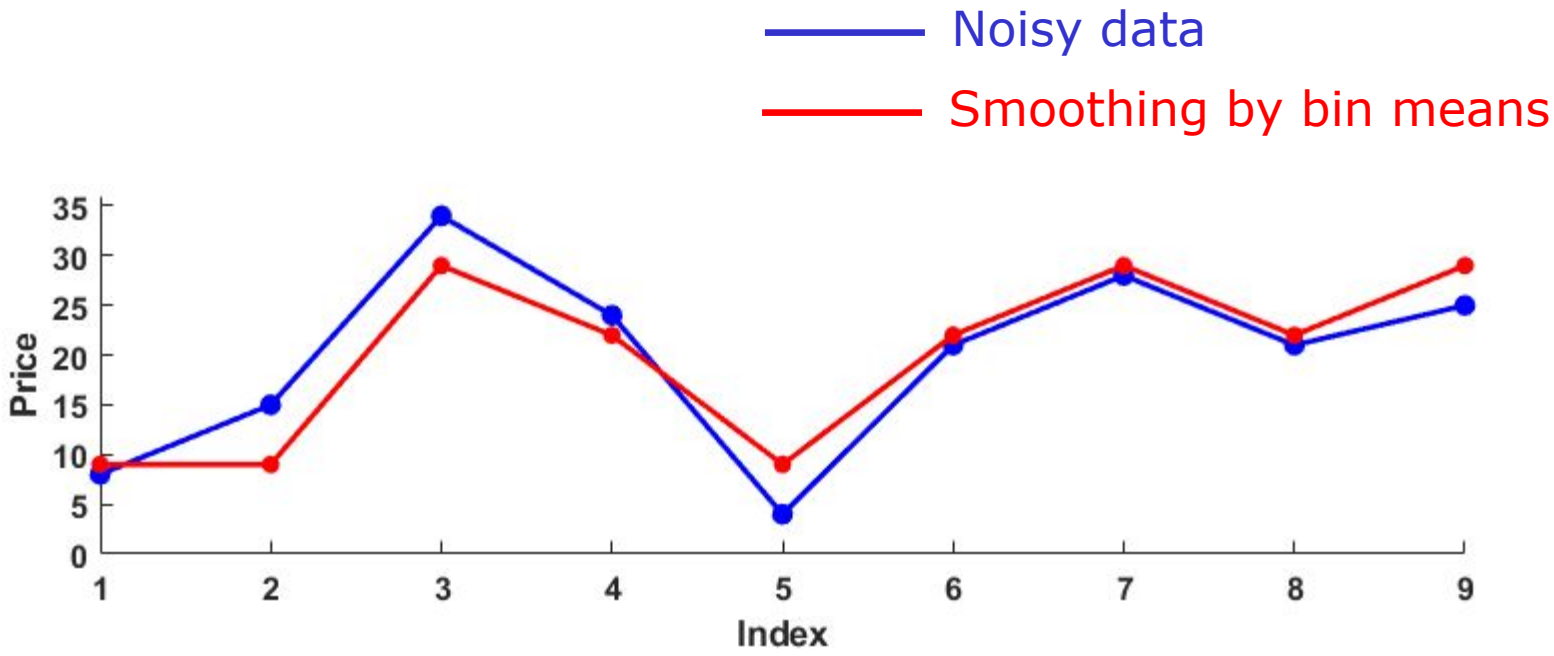


Illustration of Binning Methods for Data Smoothing

- Example:
- Noisy data for price (in Rs) : 8, 15, 34, 24, 4, 21, 28, 21, 25
- Sorted data for price (in Rs) : 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into bins:

Bin1: 4, 8, 15

Bin2: 21, 21, 24

Bin3: 25, 28, 34

Smoothing by bin Boundaries:

Bin1: 4, 4, 15

Bin2: 21, 21, 24

Bin3: 25, 25, 34

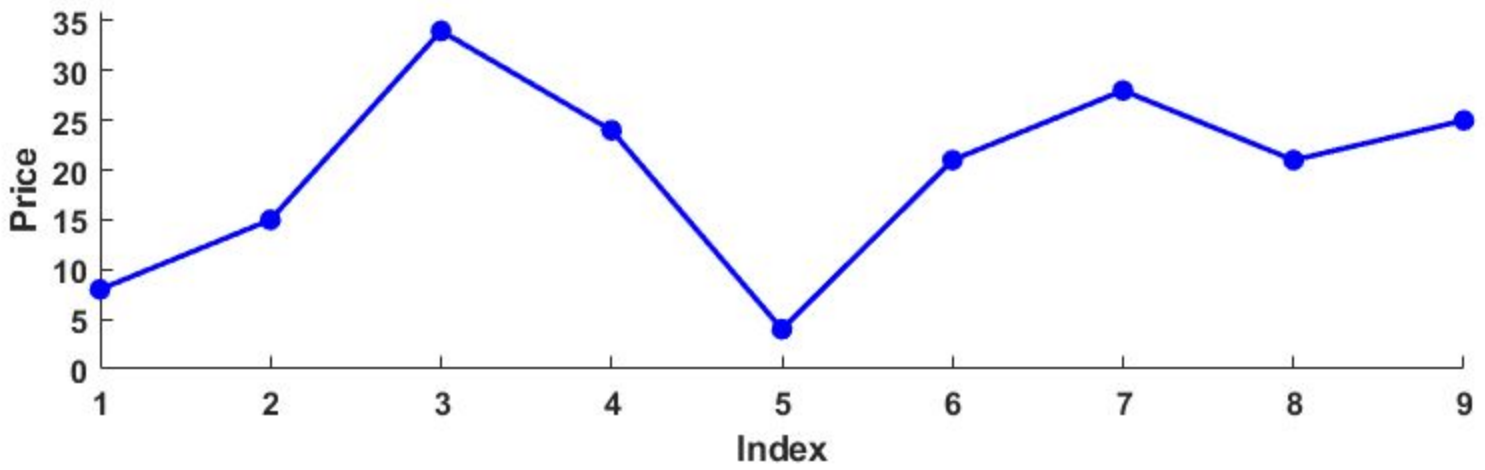


Illustration of Binning Methods for Data Smoothing

- Example:
- Noisy data for price (in Rs) : 8, 15, 34, 24, 4, 21, 28, 21, 25
- **Smoothing by bin boundaries** : 4, 15, 34, 24, 4, 21, 25, 21, 25

Partition into bins:

Bin1: 4, 8, 15

Bin2: 21, 21, 24

Bin3: 25, 28, 34

Smoothing by bin Boundaries:

Bin1: 4, 4, 15

Bin2: 21, 21, 24

Bin3: 25, 25, 34

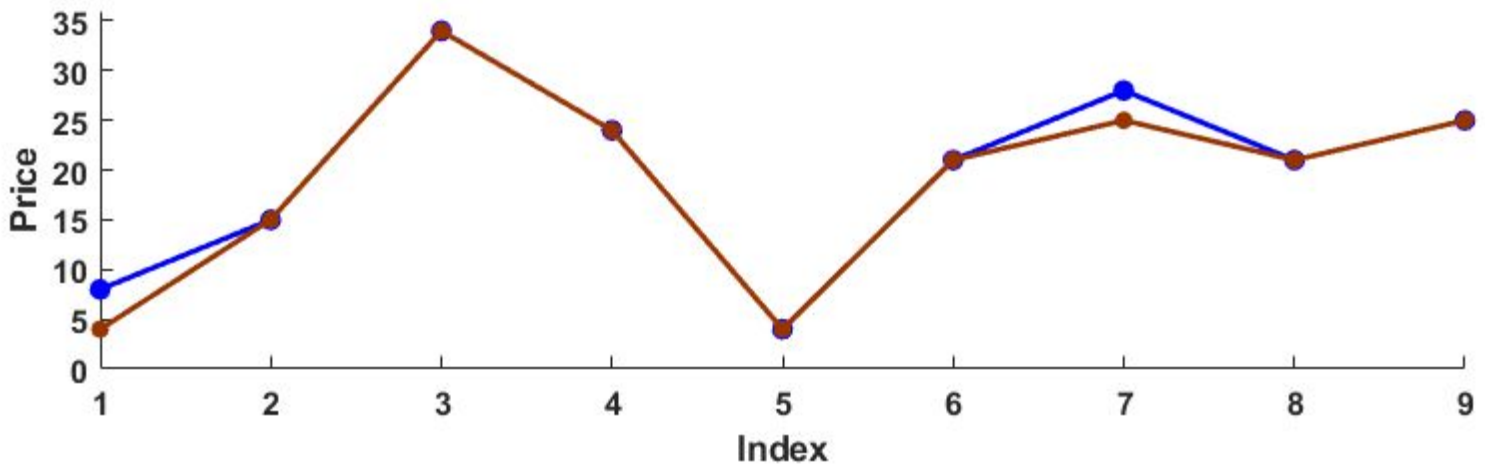
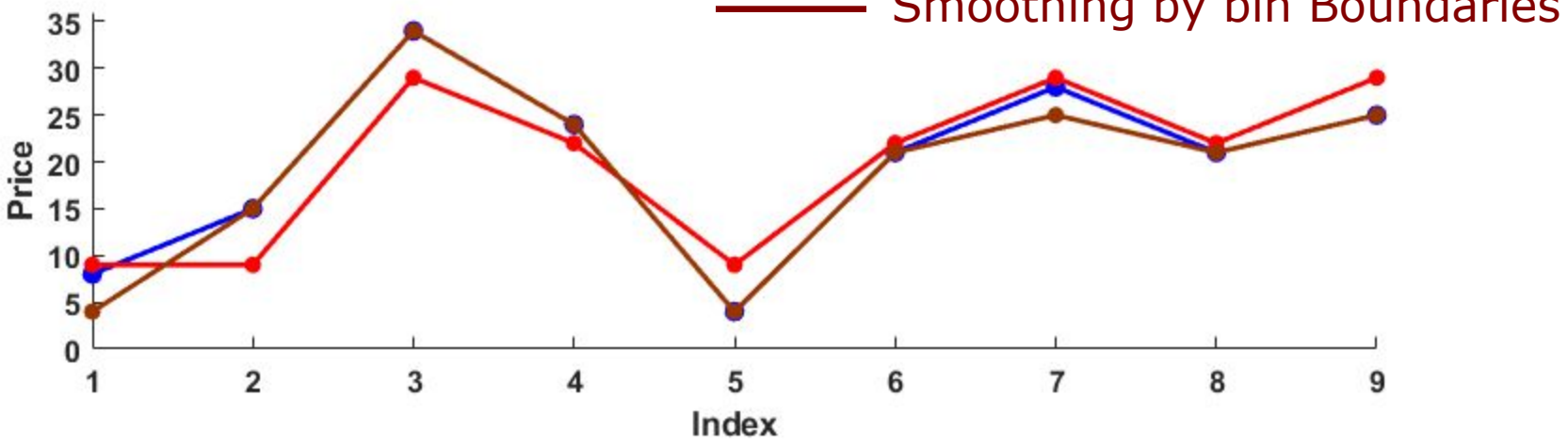


Illustration of Binning Methods for Data Smoothing

- Example:
- Noisy data for price (in Rs) : 8, 15, 34, 24, 4, 21, 28, 21, 25
- Sorted data for price (in Rs) : 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into bins:	Smoothing by bin means:	Smoothing by bin Boundaries:
Bin1: 4, 8, 15	Bin1: 9, 9, 9	Bin1: 4, 4, 15
Bin2: 21, 21, 24	Bin2: 22, 22, 22	Bin2: 21, 21, 24
Bin3: 25, 28, 34	Bin3: 29, 29, 29	Bin3: 25, 25, 34



Outlier Detection and Replacing with Centre of Tendency

- Compute first quartile (Q1) and third quartile (Q3) for an attribute
- Compute the interquartile range (IQR) as IQR=Q3-Q1 for that attribute
- Compute
  - *Lower Bound* = | Q1 – (1.5 x IQR) |
  - *Upper Bound* = | Q3 + (1.5 x IQR) |
- Detect attribute value as outlier if
  - it is less than *Lower Bound* **OR**
  - it is larger than *Upper Bound*
- Replace these outlier values with mean/median/mode of the attribute
- Important note: If the outliers are due to noise, then it is better to replace
  - Domine knowledge is very important

Summary of Data Cleaning

- 80% of data analyst's time spent in cleaning that data
- Data cleaning routines attempt to identify missing values, fill in missing values, smooth out noise while identifying outliers
- One of the biggest data cleaning task is handling missing values
- Among the different methods for filling the missing values
  - Filling by central tendency (mean/median/mode)
  - Filling by interpolation
  - Filling by regression are popular methods
- When data is mostly full of noise, smooth out the data to remove the effect of noise (binning and regression)
- Outliers can be detected using quartiles and IQR
  - Detected outliers can be replaced by mean/median/mode

Data Preprocessing
Data Integration

Data Integration

- Data integration is the process of combining the data from multiple sources into a coherent data store
- These sources may include multiple databases or flat files
- Example:
  - Temperature sensor, pressure sensor and rain gauge records temperature, atmospheric pressure and amount of rain at different locations
  - Each location has separate temperature, pressure and amount of rain tables (database)

Data Integration

- Data integration is the process of combining the data from multiple sources into a coherent data store
- These sources may include multiple databases or flat files
- Example:

	Dates	Station Id	Temperature	Humidity	Rain
2	08-07-2018	t10	25.46875	82.1875	6.75
3	09-07-2018	t10	26.19298	85.42	1762
4	10-07-2018	t10	25.17021	85.34043	652.5
5	11-07-2018	t10	25.1368	87.68657	963
6	08-07-2018	t11	23.53846	61.92308	3
7	09-07-2018	t11	26.8494	85.42	15
8	10-07-2018	t11	25.1368	75.07463	13583
9	11-07-2018	t11	27.35915	76.02113	19769
10	23-07-2018	t12	25.1368	94.4065	1071
11	24-07-2018	t12	24.16197	97.66901	438.8
12	25-07-2018	t12	25.29323	94.84211	13667
13	26-07-2018	t12	22.19718	99	864

Data Integration

- Data integration is the process of combining the data from multiple sources into a coherent data store
- These sources may include multiple databases or flat files
- Example:
  - Temperature sensor, pressure sensor and rain gauge records temperature, pressure and amount of rain at different locations
  - Each location has separate temperature, pressure and amount of rain tables (database)

- Issues to consider during data integration:
  - Schema integration (entity matching)
  - Data value conflict
  - Redundancy

Schema Integration (Entity Matching)

- Database schema: The organization of data as a blueprint of how the database is constructed
- Entity: Each entity in real-world problem is the attribute in the database
- Addresses the question of
  - "*how can equivalent real-world entities from multiple sources be matched up?*"
  - "*how can data analysts are sure that they are same?*"
- Attribute name conflict across the multiple sources of data
  - Example: `customer_id`, `customer_num`, `cust_num`
- Entity identification problem:
  - Metadata is associated with each attribute
  - Metadata include:
    - Name, Meaning, Data type, Range of values permitted

Data Value Conflict

- Issue: Detection and resolution of data value conflicts
- For the same real-world entity, attribute values from different sources may differ
- This may be due to difference in representation, scaling, or encoding
- Example:
  - "`weight`" attribute may be stored in metric unit (gram, kilogram, etc.) in one system, British imperial unit (pound, ounce, etc.) in another system
  - In a database for hotel chain in different countries:
    - "`price of room`" attribute may be stored with price value in different currencies
  - Categorical data: "`gender`" may be stored with male and female or M and F

Redundancy

- Major issue to be addressed
- Sources of redundancy:
  - An attribute may be redundant, if it can be derived from another attribute or set of attributes
    - Example: Attribute "`Total Marks`" derived from Marks from each courses
  - Inconsistency in the attribute naming can also cause redundancy in resulting data sets
    - Example: (1) `registration_id` and `roll_num`
      (2) `customer_id` and `customer_num`
- Two types of redundancies:
  - Redundancy between the attributes
  - Redundancy at the tuple level
    - Duplication of tuples
    - Remove the duplicate tuples

Redundancy Between Attributes

- Two attributed may be related or dependent
- Detected by the correlation analysis
- Correlation analysis measures how strongly one attribute implies (related) to other, based on available data
- Correlation analysis for numerical attributes:
  - Compute correlation coefficient between two attributes A and B (e.g. Pearson's product moment coefficient i.e. Pearson's correlation coefficient)
- Correlation analysis for categorical attributes:
  - Correlation (relationship) between two categorical attributes A and B can be discovered by χ^2 (chi-square) test

Redundancy Between Numerical Attributes

- Pearson's correlation coefficient ($\rho_{A,B}$):

$$\rho_{A,B} = \frac{\frac{1}{N}\sum_{i=1}^{N}(a_i - \mu_A)(b_i - \mu_B)}{\sigma_A \sigma_B} = \frac{\mathrm{Cov}(A, B)}{\sigma_A \sigma_B}$$

  - N : number of tuples
  - a_i and b_i: respective values of attribute A and attribute B in tuple i
  - μ_A and μ_B: respective mean values of A and B
  - σ_A and σ_B : respective standard deviation of A and B
  - $\mathrm{Cov}(A, B)$: Covariance between A and B

- Note: $-1 \leq \rho_{A,B} \leq +1$

Redundancy Between Numerical Attributes: Pearson's correlation coefficient

- If $\rho_{A,B}$ is greater than 0, then attributes A and B are positively correlated
  - The values of A increases as the values of B increases or vice versa
  - The higher the value, the stronger the correlation
- If $\rho_{A,B}$ is equal to 0, then attributes A and B have no correlation between them (may be independent)
- If $\rho_{A,B}$ is less than 0, then attributes A and B are negatively correlated
  - The values of A increases as the values of B decreases or vice versa
  - Each attribute discourages the other
  - The higher the value, the stronger the correlation
- A higher correlation value may indicate that A (or B) may be removed as a redundancy

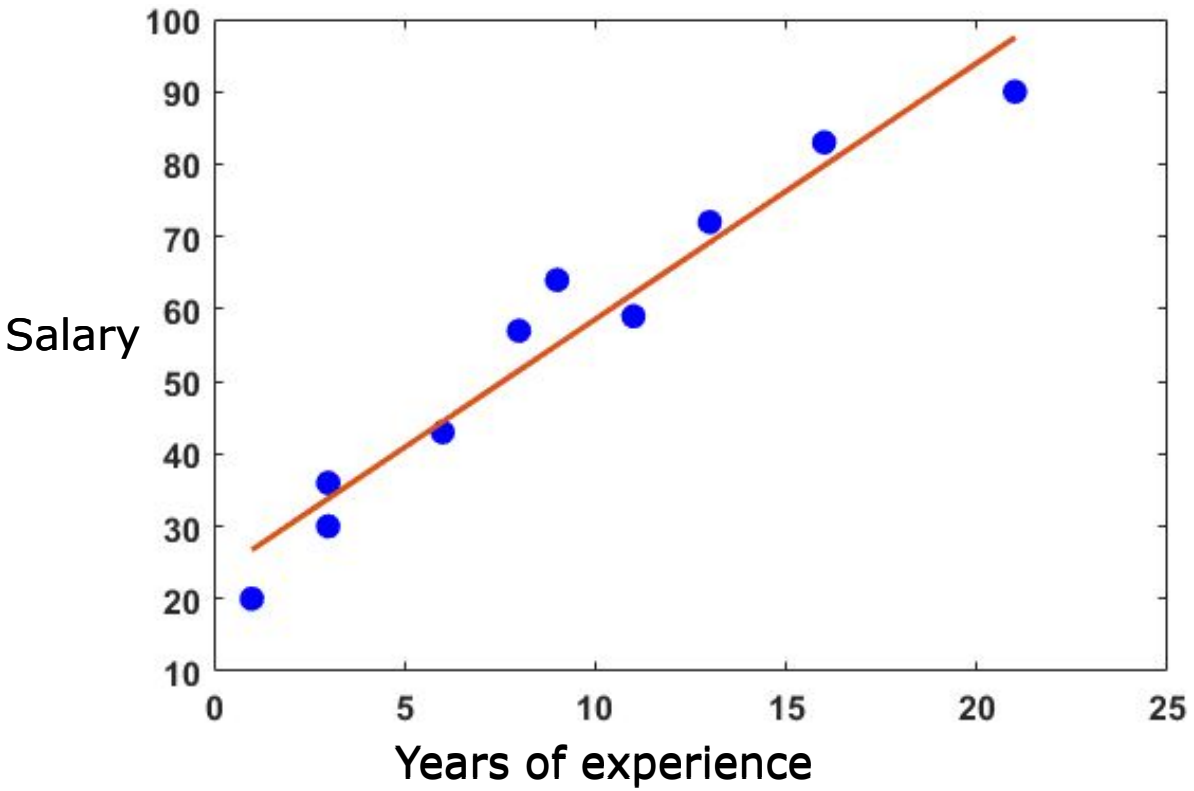
Redundancy Between Numerical Attributes: Pearson's correlation coefficient

- Assumption:
  - Both attributes (variables) should be normally distributed (normally distributed variables (normal distribution) have a bell-shaped curve)
  - Linearity: The two attributes have linear relationship
  - Homoscedasticity: Data is equally distributed about the regression line.
- Scatter plots can also be use to view correlation between the numerical attributes

Illustration of Pearson's Correlation Coefficient

Years of experience (x)	Salary (in Rs 1000) (y)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

- Scatter plot



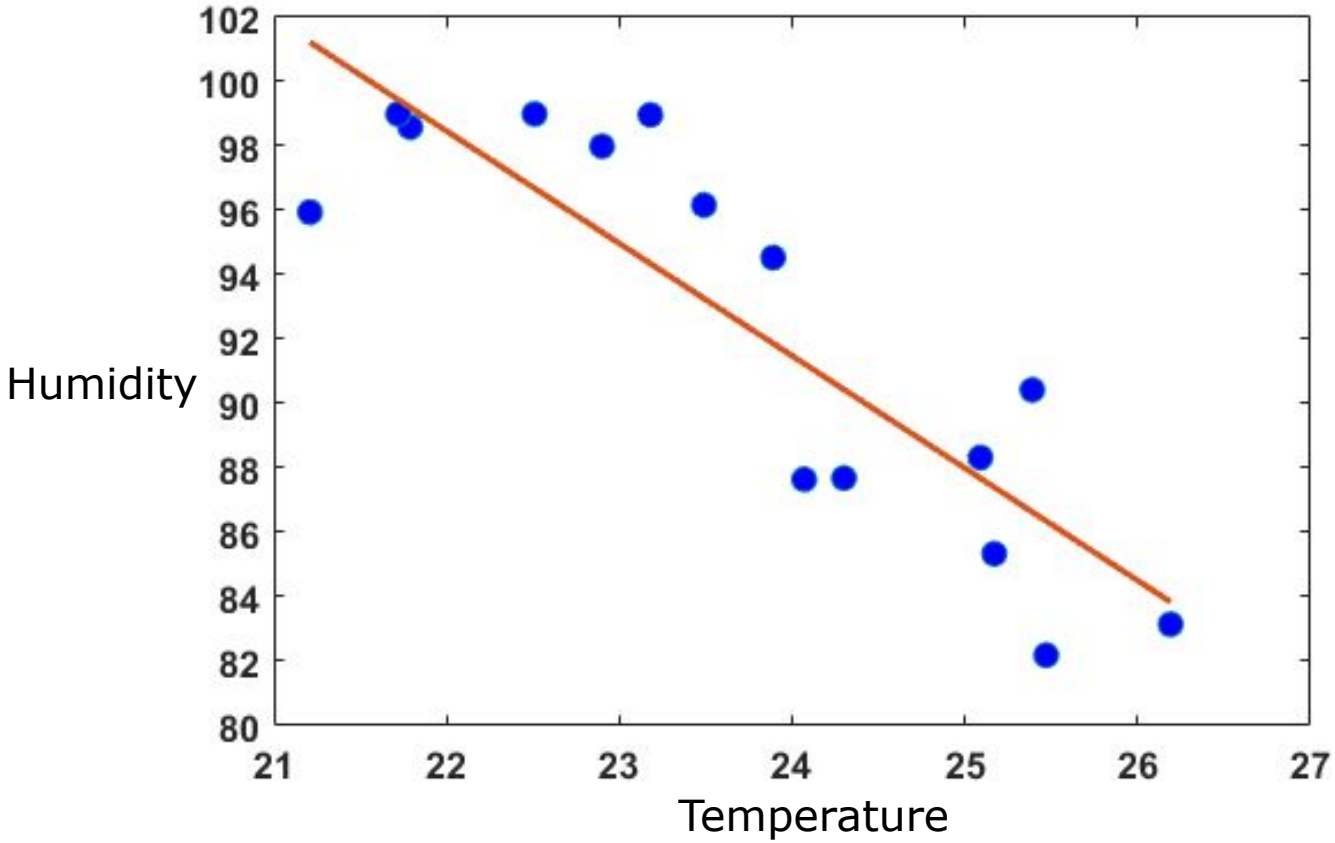
- The two attributes have linear relationship
- Data is equally distributed about the regression line (roughly)

$\rho_{A,B} = 0.97$

Illustration of Pearson's Correlation Coefficient

Temp (x)	Humidity (y)
25.47	82.19
26.19	83.15
25.17	85.34
24.30	87.69
24.07	87.65
21.21	95.95
23.49	96.17
21.79	98.59
25.09	88.33
25.39	90.43
23.89	94.54
22.51	99.00
22.90	98.00
21.72	99.00
23.18	98.97

- Scatter plot

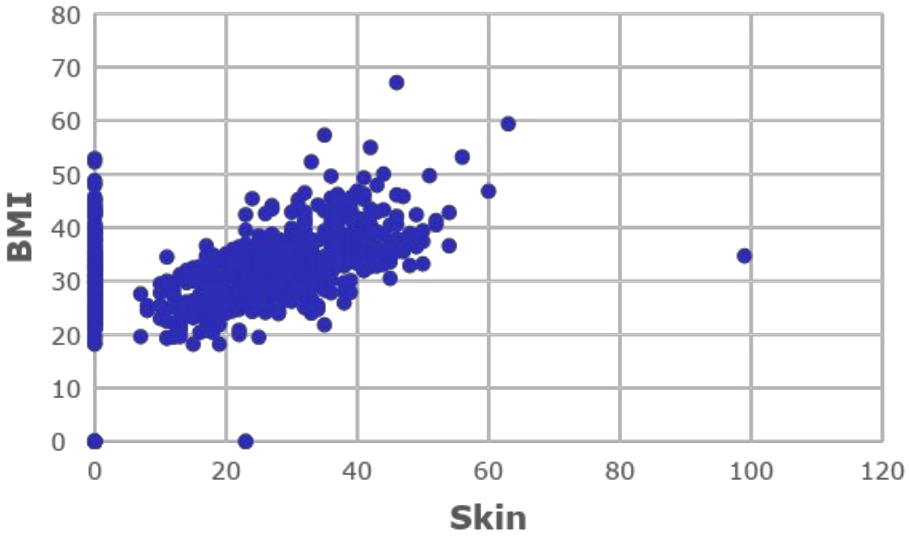


$$\rho_{A,B} = -0.8648$$

Illustration of Pearson's Correlation Coefficient: Pima-Indians-Diabetes Dataset

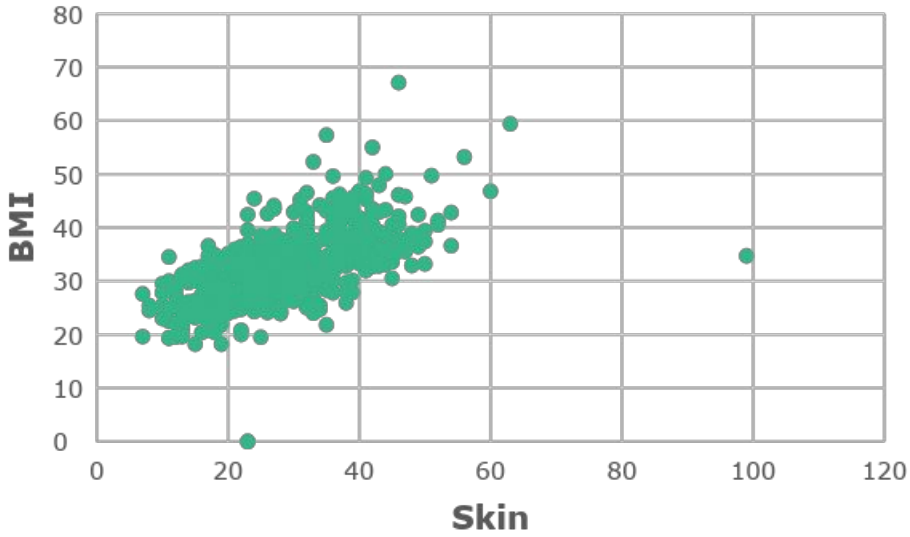
- Scatter plot between BMI (A) and Skin (B)

Before cleaning the data



$\rho_{A,B} = 0.3926$

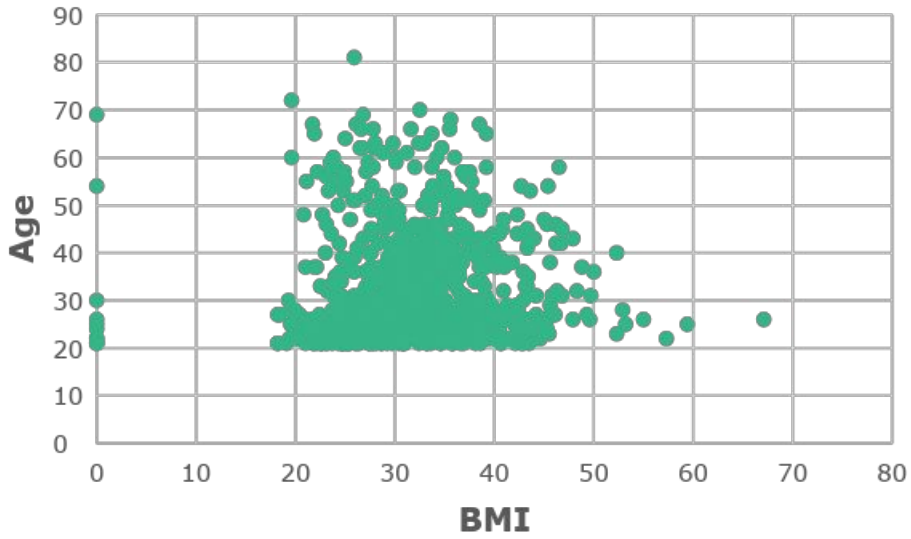
After cleaning the data



$\rho_{A,B} = 0.6320$

Illustration of Pearson's Correlation Coefficient: Pima-Indians-Diabetes Dataset

- Scatter plot between BMI (A) and Age (B)



$\rho_{A,B} = 0.0564$

Redundancy Between Numerical Attributes: Spearman Rank Correlation

- Rank correlation between variables: Statistical dependence between the rankings of two variables
  - The values the variables take should be at least ordinal
- The values in the attributes should be converted into ranks of the values (ordinal values), if the attribute is not ordinal
- Spearman rank correlation is a non-parametric measure of rank correlation between two attributes (variables)
- As it is non-parametric measure, it does not carry any assumptions about the distribution of the data
- The Spearman correlation coefficient is defined as the *Pearson correlation coefficient* between the rank variables

Redundancy Between Numerical Attributes: Spearman Rank Correlation

- Spearman correlation coefficient ($\rho_{RA,\,RB}$):

$$\rho_{R_A,R_B} = \frac{\mathrm{Cov}(R_A, R_B)}{\sigma_{R_A}\sigma_{R_B}}$$

  - R_A and R_B : ranks attribute A and attribute B
  - σ_{RA} and σ_{RB} : respective standard deviation of ranks of A and B
  - $\mathrm{Cov}(R_A, R_B)$: Covariance between the ranks of A and B

- Only if all N ranks are *distinct integers*, then it can be computed using the popular formula

$$\rho_{R_A,R_B} = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{N(N^2-1)}$$

  - N : number of tuples
  - d_i: difference between the rank values of A and B in tuple i

$$-1 \le \rho_{R_A,R_B} \le +1$$

Redundancy Between Numerical Attributes: Spearman Rank Correlation

- Pearson's correlation assesses linear relationships
- Spearman's correlation assesses monotonic relationships (whether linear or not)

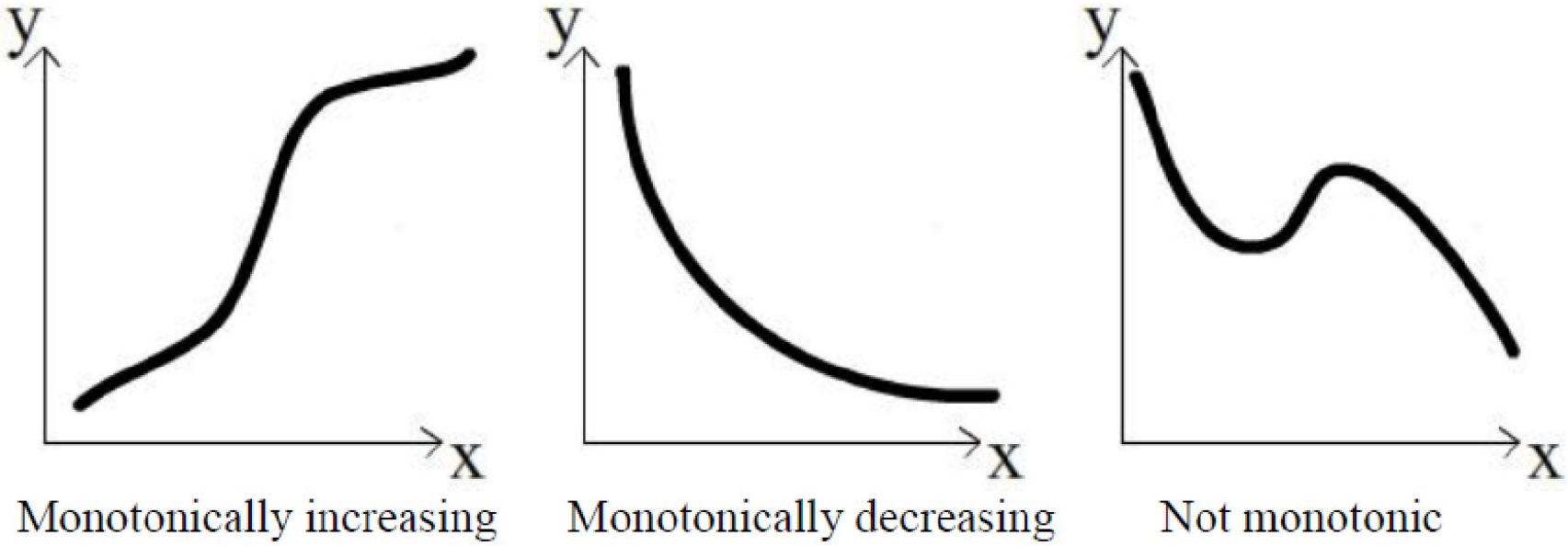


Illustration of Spearman's Correlation Coefficient

Years of experience (x)	Salary (in Rs 1000) (y)	R_x	R_y
3	30	2	2
8	57	4	5
9	64	5	7
13	72	7	8
3	36	2	3
6	43	3	4
11	59	6	6
21	90	9	10
1	20	1	1
16	83	8	9

$$\rho_{Rx,\,Ry} = 0.9806$$

- Convert the values of both attribute into rank values

Illustration of Spearman's Correlation Coefficient

Temp (x)	Humidity (y)	Rx	Ry
25.47	82.19	14	1
26.19	83.15	15	2
25.17	85.34	12	3
24.30	87.69	10	5
24.07	87.65	9	4
21.21	95.95	1	9
23.49	96.17	7	10
21.79	98.59	3	12
25.09	88.33	11	6
25.39	90.43	13	7
23.89	94.54	8	8
22.51	99.00	4	14
22.90	98.00	5	11
21.72	99.00	2	14
23.18	98.97	6	13

$$\rho_{\mathrm{R}x,\ \mathrm{R}y} = -0.8523$$

Correlation Between Numerical Attributes: Summary

- Pearson's correlation coefficient is applied on the two continuous valued attributes
- Spearman's correlation coefficient is applied on the two ranked valued (ordinal) discrete attributes

ρ range in positive correlation	ρ range in negative correlation	Correlation between A and B
0.0	0.0	None
(0.0 0.1]	(-0.0 -0.1]	Weak
(0.1 0.3]	(-0.1 -0.3]	Moderate
(0.3 0.5]	(-0.3 -0.5]	Strong
(0.5 1.0)	(-0.5 -1.0)	Very Strong
1.0	-1.0	Perfect

Redundancy Between Categorical (Discrete) Attributes

- Correlation relationship between two categorical attributes A and B can be discovered by χ^2 (chi-square) test
- Steps in χ^2 (chi-square) test :
  - Identify the two categorical attributes
  - Null hypothesis: Two attributes are independent (not related)
  - Complete the contingency matrix (table) with observed frequencies (count) and expected frequencies (probability)
  - Calculate the observed χ^2 value based on contingency matrix
  - Use the standard χ^2 table compare if the observed χ^2 value to critical χ^2 value for the problem's degree of freedom and confidence (significance i.e. p-value) level
    - If the observed χ^2 value < critical χ^2 value then the attributes are not related (null-hypothesis is true)

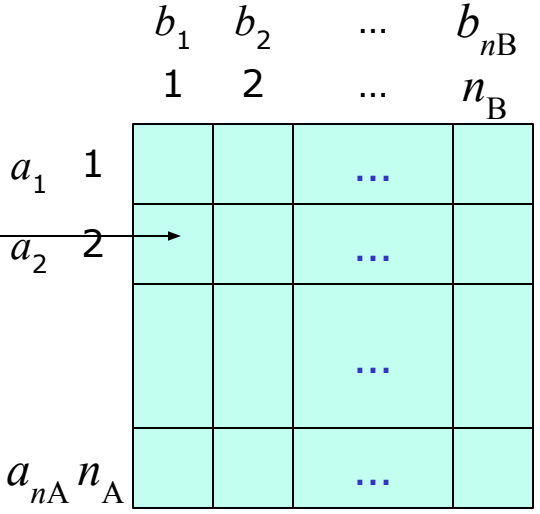
Redundancy Between Categorical (Discrete) Attributes

- Correlation relationship between two categorical attributes `A` and `B` can be discovered by χ^2 (chi-square) test
- Suppose attribute `A` has n_A distinct value $(a_1, a_2, \ldots, a_i, \ldots, a_{nA})$
- Example: Let the attribute be `gender`
  - The distinct values `gender` can take are *male* and *female*
  - Number of distinct values: 2 i.e. $n_A = 2$

Redundancy Between Categorical (Discrete) Attributes

- Correlation relationship between two categorical attributes A and B can be discovered by χ^2 (chi-square) test
- Suppose attribute A has n_A distinct value $(a_1, a_2, \ldots, a_i, \ldots, a_{nA})$
- Suppose attribute B has n_B distinct value $(b_1, b_2, \ldots, b_j, \ldots, b_{nB})$
- The data tuples described by attributes A and B can be shown as a contingency table

  - Contingency table has
    - n_A distinct values of A making up the rows
    - n_B distinct values of B making up the columns



(a_i, b_j) denote event that i^{th} district value of A and j^{th} distinct value of B taken on jointly

Redundancy Between Categorical (Discrete) Attributes

- The observed χ^2 (chi-square) value (Pearson χ^2 statistics) is computed as

$$\chi^2 = \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

  - o_{ij} : observed frequency (actual count) of joint event (A_i, B_j)
    - Number of times the i^{th} distinct value of attribute A is occurring jointly with j^{th} distinct value of attribute B
  - e_{ij} : expected frequency (probability) of joint event (A_i, B_j)
- Expected frequency (e_{ij}), i.e., probability that i^{th} distinct value of attribute A is occurring jointly with j^{th} distinct value of attribute B, is computed as

$$e_{ij} = \frac{\text{count}(A = a_i) \text{ x } \text{count}(B = b_j)}{N}$$

  - N : number of tuples
  - $\text{Count}(A = a_i)$: The number of tuple having distinct value a_i for A
  - $\text{Count}(B = b_j)$: The number of tuple having distinct value b_j for B

Redundancy Between Categorical (Discrete) Attributes

- The χ^2 statistic tests the hypothesis that A and B are independent or not related (Null hypothesis)
- The test is based on the significance level (p-value), with $(n_A\text{-}1)*(n_B\text{-}1)$ degree of freedom
  - p-value gives the evidence against the Null hypothesis
  - Smaller the p-value, stronger the evidence (confidence)
  - Example: p-value = 0.01 means 99% confidence that you can accept/reject the Null hypothesis
  - Degree of freedom (statistics): The number of values in the final calculation of a statistic that are free to vary
    - Usually one less than the number of items
- If the hypothesis can be rejected, then we say that A and B are statistically related or associated for the given data set

Redundancy Between Categorical Attributes: Illustration

- A group of 15 people are surveyed
- The gender of each person is noted
- Each person is polled as to whether their preferred type of reading material was fiction or nonfiction
- This leads to two attributes `gender` and `preferred_reading`
  - `gender` takes two district values male and female
  - `preferred_reading` takes two distinct values fiction and non-fiction

Redundancy Between Categorical Attributes: Illustration

Sl. No.	gender	preferred_reading
1	male	fiction
2	female	Non-fiction
3	female	Non-fiction
4	male	Non-fiction
5	female	fiction
6	female	Non-fiction
7	male	Non-fiction
8	male	fiction
9	female	Non-fiction
10	female	fiction
11	male	fiction
12	female	Non-fiction
13	female	Non-fiction
14	male	fiction
15	male	Non-fiction

- A group of 15 people are surveyed
- Size of the contingency matrix is 2 x 2

	fiction (b_1)	Non-fiction (b_2)	Total
male (a_1)	4 (o_{11})	3 (o_{12})	7
female (a_2)	2 (o_{21})	6 (o_{22})	8
Total	6	9	15

Redundancy Between Categorical Attributes: Illustration

Sl. No.	gender	preferred_reading
1	male	fiction
2	female	Non-fiction
3	female	Non-fiction
4	male	Non-fiction
5	female	fiction
6	female	Non-fiction
7	male	Non-fiction
8	male	fiction
9	female	Non-fiction
10	female	fiction
11	male	Non-fiction
12	female	Non-fiction
13	female	Non-fiction
14	male	fiction
15	male	Non-fiction

- A group of 15 people are surveyed
- Size of the contingency matrix is 2 x 2

	fiction (b_1)	Non-fiction (b_2)	Total
male (a_1)	4 (o_{11}) 2.8 (e_{11})	3 (o_{12}) 4.2 (e_{12})	7
female (a_2)	2 (o_{21}) 3.2 (e_{21})	6 (o_{22}) 4.8 (e_{22})	8
Total	6	9	15

$$e_{11} = \frac{\text{count(male) x count(fiction)}}{N} = \frac{7 \text{ x } 6}{15} = 2.8$$

$$e_{12} = \frac{\text{count(male) x count(nonfiction)}}{N} = \frac{7 \text{ x } 9}{15} = 4.2$$

$$e_{21} = \frac{\text{count(female) x count(fiction)}}{N} = \frac{8 \text{ x } 6}{15} = 3.2$$

$$e_{22} = \frac{\text{count(female) x count(nonfiction)}}{N} = \frac{8 \text{ x } 9}{15} = 4.8$$

Redundancy Between Categorical Attributes: Illustration

	fiction (b_1)	Non-fiction (b_2)	Total
male (a_1)	4 (o_{11}) 2.8 (e_{11})	3 (o_{12}) 4.2 (e_{12})	7
female (a_2)	2 (o_{21}) 3.2 (e_{21})	6 (o_{22}) 4.8 (e_{22})	8
Total	6	9	15

- The numbers in blue are the expected frequencies (probability)
- The χ^2 value is computed as

$$\chi^2 = \frac{(o_{11}-e_{11})^2}{e_{11}} + \frac{(o_{12}-e_{12})^2}{e_{12}} + \frac{(o_{21}-e_{21})^2}{e_{21}} + \frac{(o_{22}-e_{22})^2}{e_{22}}$$

$$\chi^2 = \frac{(4-2.8)^2}{2.8} + \frac{(3-4.2)^2}{4.2} + \frac{(2-3.2)^2}{3.2} + \frac{(6-4.8)^2}{4.8}$$

$$\chi^2 = 1.607$$

Redundancy Between Categorical Attributes: Illustration

- For 2 x 2 contingency table, the degree of freedom is (2-1)*(2-1) = 1
- Obtain the χ^2 value for 0.05 significance i.e. p=0.05 (95% chance or confidence) with 1 degree of freedom
  - χ^2 value is 3.841 (Taken from the table of χ^2 distribution)
- Computed χ^2 value for given population is 1.607
- The computed value is less than the 3.841
  - We accept the hypothesis that `gender` and `preferred_reading` are independent (not related)
- **Conclusion**: The two attributes (`gender` and `preferred_reading`) are not correlated for the given group of people

Data Preprocessing

Data Transformation

Data Transformation

- The data are transformed or consolidated into the forms appropriate for data modelling using machine learning
- Data Transformation involve
  - Smoothing:
    - Used for removing noise or reducing the effect of noise
    - Techniques: Binning, Regression, Clustering
  - Aggregation:
    - Summery or aggregation operation are applied to the data
    - Analysis of data at multiple granularity
      - Example: Daily sales data, Monthly sales data (aggregated on daily data)
  - Attribute construction (feature construction):
    - New attributes are constructed from the raw-data to help mining process
  - Normalization and standardization

Attribute Normalization

- In the context of machine learning, it is termed as feature normalization
- An attribute is normalised by scaling its value so that they fall within a small specified range (for example 0.0 to 1.0)
- Normalization is particularly useful for classification algorithms involving distance measurements and clustering
- For distance based approaches, normalization helps prevent attributes with large ranges from overweighting attributes with smaller ranges

Illustration

	x_1	x_2
Salesman-ID	**Total sales (Rs)**	**Score for sale**
S001	23500.00	8
S002	23500.00	6
S003	22879.00	2
S004	2300.00	4
S005	34678.00	5
S006	15687.00	8
S007	18945.00	8
S008	8750.00	2
S009	37489.00	4
S010	73567.00	2
S011	52789.00	4
S012	2900.00	3
S013	6570	3
S014	21000.00	2
min:	**2300.00**	**2**
max:	**73567.00**	**8**

y_1	y_2
23000.00	6.5

$$\text{Eucledin Distance (ED)} = \sum_{i=1}^{d} (x_i - y_i)^2$$

ED1 = $(23500.00 - 23000.00)^2 + (8 - 6.5)^2$

ED1 = **250002.25**

Illustration

	x_1	x_2
Salesman-ID	**Total sales (Rs)**	**Score for sale**
S001	23500.00	8
S002	23500.00	6
S003	22879.00	2
S004	2300.00	4
S005	34678.00	5
S006	15687.00	8
S007	18945.00	8
S008	8750.00	2
S009	37489.00	4
S010	73567.00	2
S011	52789.00	4
S012	2900.00	3
S013	6570	3
S014	21000.00	2
min:	**2300.00**	**2**
max:	**73567.00**	**8**

y_1	y_2
23000.00	6.5

$$\text{Eucledin Distance (ED)} = \sum_{i=1}^{d} (x_i - y_i)^2$$

ED1 = $(23500.00 - 23000.00)^2 + (8 - 6.5)^2$
ED1 = **250002.25**

ED1 = $(23500.00 - 23000.00)^2 + (6 - 6.5)^2$
ED1 = **250000.25**

Illustration

	x_1	x_2
Salesman-ID	Total sales (Rs)	Score for sale
S001	23500.00	8
S002	23500.00	6
S003	22879.00	2
S004	2300.00	4
S005	34678.00	5
S006	15687.00	8
S007	18945.00	8
S008	8750.00	2
S009	37489.00	4
S010	73567.00	2
S011	52789.00	4
S012	2900.00	3
S013	6570	3
S014	21000.00	2
min:	**2300.00**	**2**
max:	**73567.00**	**8**

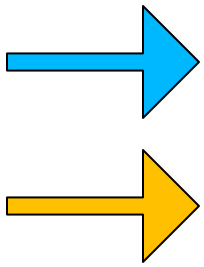
y_1	y_2
23000.00	6.5

$$\text{Eucledin Distance (ED)} = \sum_{i=1}^{d} (x_i - y_i)^2$$

ED1 = $(23500.00 - 23000.00)^2 + (8 - 6.5)^2$
ED1 = **250002.25**

ED1 = $(23500.00 - 23000.00)^2 + (6 - 6.5)^2$
ED1 = **250000.25**

ED3 = $(22879.00 - 23000.00)^2 + (2 - 6.5)^2$
ED3 = **14661.25**



Attribute Normalization: Min-Max Normalization

- It performs a linear transformation on the original data
- The transformed data is the scaled version of the original data so that they fall within a small specified range
- Each numeric attributes in a data are normalised separately
- Steps:
  - Compute minimum (mn_A) and maximum (mx_A) values of an attribute A
  - Specify the new minimum (new_mn_A) and new maximum range (new_mx_A)
  - Min-Max normalization maps a value, x of attribute A to $\hat{x}$ in the specified range by computing

$$\hat{x} = \frac{x - mn_A}{mx_A - mn_A}(new\_mx_A - new\_mn_A) + new\_mn_A$$

Attribute Normalization: Min-Max Normalization

- When new minimum (new_mn_A) and new maximum range (new_mx_A) is 0 and 1 respectively, then the data is scaled to 0.0 to 1.0 range
  - Min-Max normalization maps a value, x of attribute A to $\hat{x}$ in the specified range by computing

$$\hat{x} = \frac{x - mn_A}{mx_A - mn_A}$$

Min-Max Normalization during Model Building

- Model building and prediction using machine learning involve two stages:
  - Training stage: Model building
  - Test stage: Prediction using the built model
- Training stage: Normalise each attribute using Min-Max normalization by using the minimum and maximum values from respective attributes
- Test stage: Normalise each test records (samples) using the minimum and maximum values from respective attributes obtained during training stage

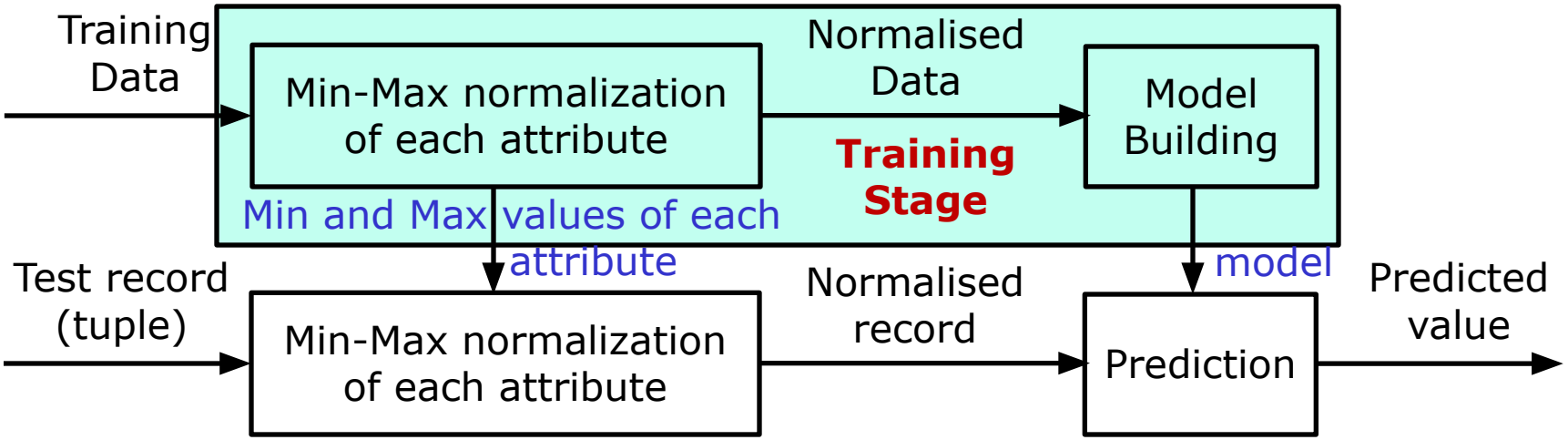
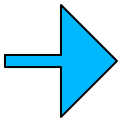


Illustration of Min-Max Normalization

	Temperature	Humidity	Rain
2	25.46875	82.1875	6.75
3	26.19298	83.14912	1762
4	25.17021	85.34043	653
5	24.29851	87.68657	963
6	24.06923	87.64615	254
7	21.20779	95.94805	340
8	23.48571	96.17143	38.3
9	21.79487	98.58974	29.3
10	25.09346	88.3271	4.5
11	25.39423	90.43269	113
12	23.89076	94.53782	736
13	22.5098	99	608
14	22.904	98	718
15	21.72464	99	513

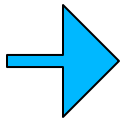


Temperature	Humidity	Rain
0.85472	0.00000	0.00128
1.00000	0.05720	1.00000
0.79484	0.18753	0.36876
0.61998	0.32708	0.54545
0.57399	0.32468	0.14213
0.00000	0.81847	0.19078
0.45694	0.83176	0.01921
0.11776	0.97560	0.01408
0.77944	0.36518	0.00000
0.83978	0.49042	0.06146
0.53819	0.73459	0.41613
0.26118	1.00000	0.34315
0.34025	0.94052	0.40589
0.10368	1.00000	0.28937

min:	21.20779	82.187	4.5	0.000	0.000	0.000
max:	26.19298	99	1762	1.000	1.000	1.000

Illustration of Min-Max Normalization

Salesman-ID	Total sales (Rs)	Score for sale
S001	23500.00	8
S002	23500.00	6
S003	22879.00	2
S004	2300.00	4
S005	34678.00	5
S006	15687.00	8
S007	18945.00	8
S008	8750.00	2
S009	37489.00	4
S010	73567.00	2
S011	52789.00	4
S012	2900.00	3
S013	6570	3
S014	21000.00	2
min:	**2300.00**	**2**
max:	**73567.00**	**8**



Salesman-ID	Total sales (Rs)	Score for sale
S001	0.2975	1.0000
S002	0.2975	0.6667
S003	0.2888	0.0000
S004	0.0000	0.3333
S005	0.4543	0.5000
S006	0.1878	1.0000
S007	0.2336	0.6667
S008	0.0905	0.0000
S009	0.4938	0.3333
S010	1.0000	0.0000
S011	0.7084	0.3333
S012	0.0084	0.1667
S013	0.0599	0.1667
S014	0.2624	0.0000
	0.0000	**0.0000**
	1.0000	**1.0000**

Illustration of Min-Max Normalization

Salesman-ID	Total sales (Rs)	Score for sale
S001	0.2975	1.0000
S002	0.2975	0.6667
S003	0.2888	0.0000
S004	0.0000	0.3333
S005	0.4543	0.5000
S006	0.1878	1.0000
S007	0.2336	0.6667
S008	0.0905	0.0000
S009	0.4938	0.3333
S010	1.0000	0.0000
S011	0.7084	0.3333
S012	0.0084	0.1667
S013	0.0599	0.1667
S014	0.2624	0.0000
min:	**0.0000**	**0.0000**
max:	**1.0000**	**1.0000**

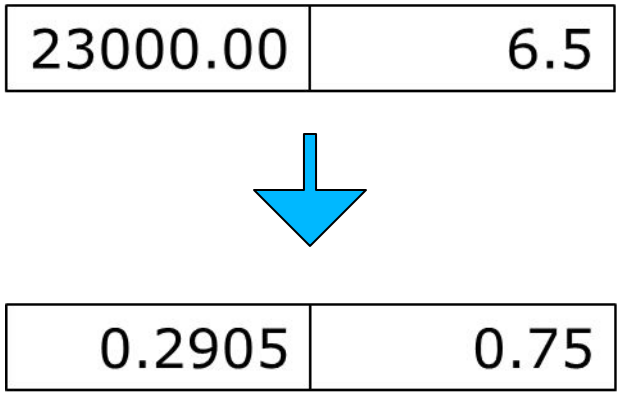


Illustration of Min-Max Normalization

Salesman-ID	Total sales (x_1) (Rs)	Score for sale (x_2)
S001	0.2975	1.0000
S002	0.2975	0.6667
S003	0.2888	0.0000
S004	0.0000	0.3333
S005	0.4543	0.5000
S006	0.1878	1.0000
S007	0.2336	0.6667
S008	0.0905	0.0000
S009	0.4938	0.3333
S010	1.0000	0.0000
S011	0.7084	0.3333
S012	0.0084	0.1667
S013	0.0599	0.1667
S014	0.2624	0.0000
min:	**0.0000**	**0.0000**
max:	**1.0000**	**1.0000**

y_1	y_2
0.2905	0.75

$$\text{Eucledin Distance (ED)} = \sum_{i=1}^{d} (x_i - y_i)^2$$

ED1 = $(0.2975 - 0.2905)^2 + (1 - 0.75)^2$

ED1 = **0.06255**

Illustration of Min-Max Normalization

	x_1	x_2
Salesman-ID	**Total sales (Rs)**	**Score for sale**
S001	0.2975	1.0000
S002	0.2975	0.6667
S003	0.2888	0.0000
S004	0.0000	0.3333
S005	0.4543	0.5000
S006	0.1878	1.0000
S007	0.2336	0.6667
S008	0.0905	0.0000
S009	0.4938	0.3333
S010	1.0000	0.0000
S011	0.7084	0.3333
S012	0.0084	0.1667
S013	0.0599	0.1667
S014	0.2624	0.0000
min:	**0.0000**	**0.0000**
max:	**1.0000**	**1.0000**

y_1	y_2
0.2905	0.75

$$\text{Eucledin Distance (ED)} = \sum_{i=1}^{d} (x_i - y_i)^2$$

ED1 = $(0.2975 - 0.2905)^2 + (1 - 0.75)^2$
ED1 = **0.06255**

ED2 = $(0.2975 - 0.2905)^2 + (0.6667 - 0.75)^2$
ED2 = **0.00699**

Illustration of Min-Max Normalization

	x_1	x_2
Salesman-ID	**Total sales (Rs)**	**Score for sale**
S001	0.2975	1.0000
S002	0.2975	0.6667
S003	0.2888	0.0000
S004	0.0000	0.3333
S005	0.4543	0.5000
S006	0.1878	1.0000
S007	0.2336	0.6667
S008	0.0905	0.0000
S009	0.4938	0.3333
S010	1.0000	0.0000
S011	0.7084	0.3333
S012	0.0084	0.1667
S013	0.0599	0.1667
S014	0.2624	0.0000
min:	**0.0000**	**0.0000**
max:	**1.0000**	**1.0000**

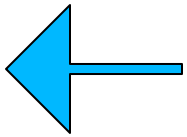
y_1	y_2
0.2905	0.75

$$\text{Eucledin Distance (ED)} = \sum_{i=1}^{d} (x_i - y_i)^2$$

ED1 = $(0.2975 - 0.2905)^2 + (1.0 - 0.75)^2$
ED1 = **0.06255**

ED2 = $(0.2975 - 0.2905)^2 + (0.6667 - 0.75)^2$
ED2 = **0.00699**

ED3 = $(0.2888 - 0.2905)^2 + (0.0 - 0.75)^2$
ED3 = **0.56250**



Attribute Normalization: Min-Max Normalization

- Min-Max normalization preserves the relationship among the original data values
- It is useful when data has varying ranges among attributes
- It is useful when machine learning (ML) algorithms we are using does not make any assumption about distribution of data
- It is useful when the actual minimum and maximum values for the attribute is known
- Disadvantage: "out-of-bound" error if a future input case for normalization falls outside the original range of attribute A

– This situation arises when the actual minimum and maximum of attribute A is unknown

Data Standardization (z-score Normalization)

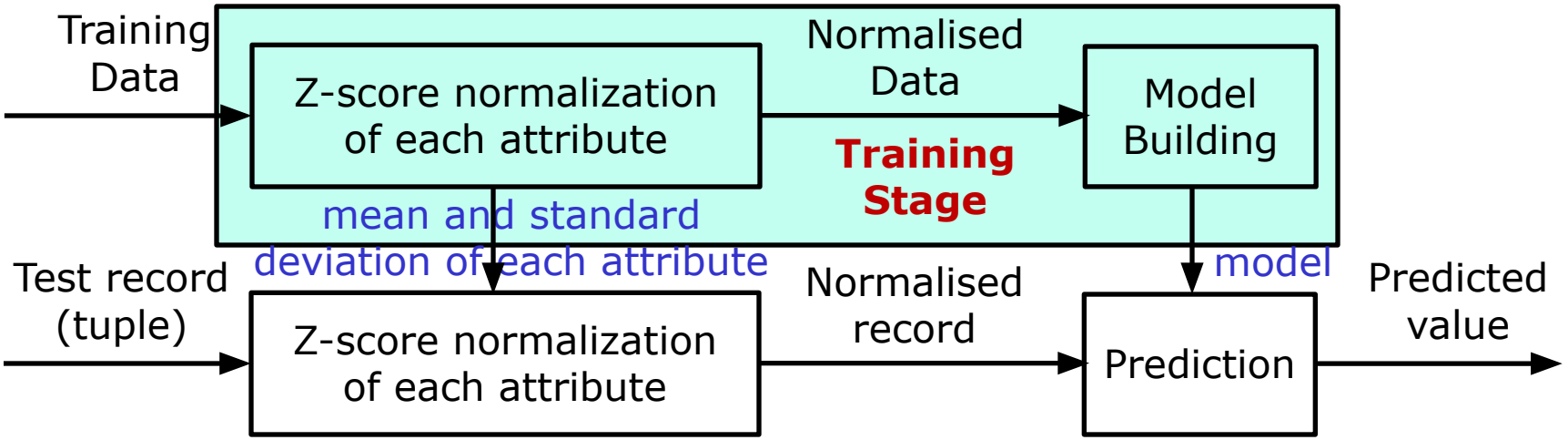
- The process of rescaling one or more attributes so that the transformed data have 0 mean and unit variance i.e. standard deviation of 1
- Standardization assumes that data is coming from Gaussian distribution
  - This assumption does not strictly have to be true, but this technique is more effective if your attribute distribution is Gaussian
- In this process, values of an attribute, A, are normalised based on the mean and standard deviation of A
  - Min-Max normalization maps a value, x of attribute A to $\hat{x}$ in the specified range by computing

$$\hat{x} = \frac{x - \mu_A}{\sigma_A}$$

- μ_A: mean of attribute A
- σ_A: standard deviation of attribute A

z-score Normalization during Model Building

- Model building and prediction using machine learning involve two stages:
  - Training stage: Model building
  - Test stage: Prediction using the built model
- Training stage: Normalise each attribute using z-score normalization by using the mean and standard deviation from respective attributes
- Test stage: Normalise each test records (samples) using the mean and standard deviation from respective attributes obtained during training stage

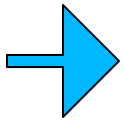


Data Standardization (z-score Normalization)

- This method of normalization is useful
  - when the actual minimum and maximum of attribute are unknown
  - when there are outliers that dominates the Min-Max normalization
  - when data follows Gaussian distribution (symmetric distribution)
- This method of normalization is useful when the ML algorithms make any assumptions of Gaussian distribution

Illustration of Data Standardization (z-score Normalization)

	Temperature	Humidity	Rain
2	25.46875	82.1875	6.75
3	26.19298	83.14912	1762
4	25.17021	85.34043	653
5	24.29851	87.68657	963
6	24.06923	87.64615	254
7	21.20779	95.94805	340
8	23.48571	96.17143	38.3
9	21.79487	98.58974	29.3
10	25.09346	88.3271	4.5
11	25.39423	90.43269	113
12	23.89076	94.53782	736
13	22.5098	99	608
14	22.904	98	718
15	21.72464	99	513

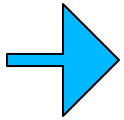


Temperature	Humidity	Rain
1.05444	-1.57673	-0.97166
1.51216	-1.41995	2.62269
0.86576	-1.06268	0.35088
0.31484	-0.68016	0.98680
0.16993	-0.68675	-0.46476
-1.63853	0.66679	-0.28965
-0.19886	0.70321	-0.90714
-1.26749	1.09749	-0.92558
0.81726	-0.57573	-0.97627
1.00735	-0.23244	-0.75508
0.05714	0.43686	0.52138
-0.81564	1.16438	0.25871
-0.56650	1.00134	0.48451
-1.31187	1.16438	0.06517

μ:	23.80035	91.86	481	0.000	0.000	0.000
σ:	1.58225	6.13	488	1	1	1

Illustration of Data Standardization (z-score Normalization)

Salesman-ID	Total sales (Rs)	Score for sale
S001	23500.00	8
S002	23500.00	6
S003	22879.00	2
S004	2300.00	4
S005	34678.00	5
S006	15687.00	8
S007	18945.00	8
S008	8750.00	2
S009	37489.00	4
S010	73567.00	2
S011	52789.00	4
S012	2900.00	3
S013	6570	3
S014	21000.00	2
μ :	**24611.00**	**4.36**
σ :	**19873.30**	**2.31**



Salesman-ID	Total sales (Rs)	Score for sale
S001	-0.06	1.58
S002	-0.06	0.71
S003	-0.09	-1.02
S004	-1.12	-0.15
S005	0.51	0.28
S006	-0.45	1.58
S007	-0.29	1.58
S008	-0.80	-1.02
S009	0.65	-0.15
S010	2.46	-1.02
S011	1.42	-0.15
S012	-1.09	-0.59
S013	-0.91	-0.59
S014	-0.18	-1.02
	0.00	**0.00**
	1.00	**1.00**

Illustration of Data Standardization (z-score Normalization)

Salesman-ID	Total sales (Rs)	Score for sale
S001	-0.06	1.58
S002	-0.06	0.71
S003	-0.09	-1.02
S004	-1.12	-0.15
S005	0.51	0.28
S006	-0.45	1.58
S007	-0.29	1.58
S008	-0.80	-1.02
S009	0.65	-0.15
S010	2.46	-1.02
S011	1.42	-0.15
S012	-1.09	-0.59
S013	-0.91	-0.59
S014	-0.18	-1.02
μ :	**0.00**	**0.00**
σ :	**1.00**	**1.00**

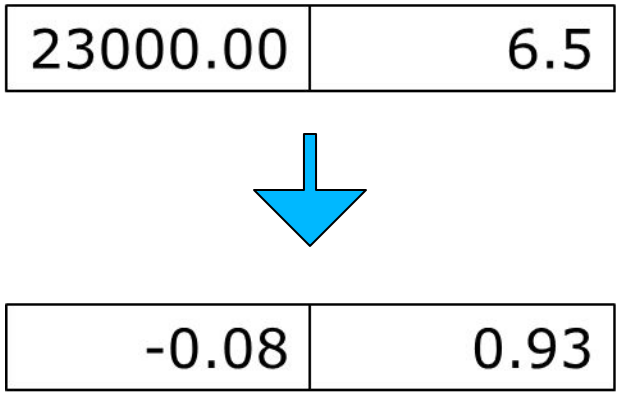


Illustration of Data Standardization (z-score Normalization)

	x_1	x_2
Salesman-ID	**Total sales (Rs)**	**Score for sale**
S001	-0.06	1.58
S002	-0.06	0.71
S003	-0.09	-1.02
S004	-1.12	-0.15
S005	0.51	0.28
S006	-0.45	1.58
S007	-0.29	1.58
S008	-0.80	-1.02
S009	0.65	-0.15
S010	2.46	-1.02
S011	1.42	-0.15
S012	-1.09	-0.59
S013	-0.91	-0.59
S014	-0.18	-1.02
μ :	**0.00**	**0.00**
σ :	**1.00**	**1.00**

y_1	y_2
-0.08	0.93

$$\text{Eucledin Distance (ED)} = \sum_{i=1}^{d} (x_i - y_i)^2$$

ED1 = $(-0.06 + 0.08)^2 + (1.58 - 0.93)^2$

ED1 = **0.42**

Illustration of Data Standardization (z-score Normalization)

Salesman-ID	Total sales (Rs) x_1	Score for sale x_2
S001	-0.06	1.58
S002	-0.06	0.71
S003	-0.09	-1.02
S004	-1.12	-0.15
S005	0.51	0.28
S006	-0.45	1.58
S007	-0.29	1.58
S008	-0.80	-1.02
S009	0.65	-0.15
S010	2.46	-1.02
S011	1.42	-0.15
S012	-1.09	-0.59
S013	-0.91	-0.59
S014	-0.18	-1.02
μ :	**0.00**	**0.00**
σ :	**1.00**	**1.00**

y_1	y_2
-0.08	0.93

$$\text{Eucledin Distance (ED)} = \sum_{i=1}^{d} (x_i - y_i)^2$$

ED1 = $(-0.06 + 0.08)^2 + (1.58 - 0.93)^2$
ED1 = **0.42**

ED2 = $(-0.06 + 0.08)^2 + (0.71 - 0.93)^2$
ED2 = **0.05**

Illustration of Data Standardization (z-score Normalization)

	x_1	x_2
Salesman-ID	**Total sales (Rs)**	**Score for sale**
S001	-0.06	1.58
S002	-0.06	0.71
S003	-0.09	-1.02
S004	-1.12	-0.15
S005	0.51	0.28
S006	-0.45	1.58
S007	-0.29	1.58
S008	-0.80	-1.02
S009	0.65	-0.15
S010	2.46	-1.02
S011	1.42	-0.15
S012	-1.09	-0.59
S013	-0.91	-0.59
S014	-0.18	-1.02
μ :	**0.00**	**0.00**
σ :	**1.00**	**1.00**

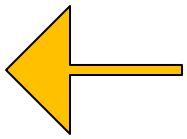
y_1	y_2
-0.08	0.93

$$\text{Eucledin Distance (ED)} = \sum_{i=1}^{d} (x_i - y_i)^2$$

ED1 = $(-0.06 + 0.08)^2 + (1.58 - 0.93)^2$
ED1 = **0.42**

ED2 = $(-0.06 + 0.08)^2 + (0.71 - 0.93)^2$
ED2 = **0.05**

ED3 = $(-0.09 + 0.08)^2 + (-1.02 - 0.93)^2$
ED3 = **3.80**



Summary on Data Transformation

- Data transformation is useful of data modelling
- Normalization:
  - Each attribute is normalised by scaling its value so that they fall within a small specified range (for example 0.0 to 1.0)
  - Min-Max normalization
    - It is useful when data has varying ranges among attributes
- Standardization (z-score normalization):
  - The process of rescaling one or more attributes so that the transformed data have 0 mean and unit variance i.e. standard deviation of 1
  - Standardization assumes that data follows a Gaussian distribution
  - It is useful when the actual minimum and maximum of attribute are unknown

Data Preprocessing

Data Reduction

Data Reduction

- Data reduction techniques are applied to obtain a reduced representation of the dataset that is much smaller in volume, yet closely maintain the integrity of the original data
- The pattern mining on the reduced dataset should produce the same or almost same analytical results
- Different strategies:
  - Attribute subset selection (feature selection):
    - Irrelevant, weekly relevant or redundant attributes (dimensions) are detected and removed
  - Dimensionality reduction:
    - Encoding mechanisms are used to reduce the dataset size

Attribute (Feature) Subset Section

- In the context of machine learning, it is termed as feature subset selection
- Irrelevant or redundant features are detected using correlation analysis
- Two strategies:
  - First strategy:
    - Perform the correlation analysis between every pair of attributes
    - Drop one among the two attributes when they are highly correlated
  - Second strategy:
    - Perform the correlation analysis between each attribute and target attribute
      - Classification: Target attribute is class label attribute
      - Regression: Target attribute is attribute corresponding to output variable
    - Drop the attributes that are less correlated with target attribute.

Attribute (Feature) Subset Section

Temperature	Humidity	Pressure	Rain
25.47	82.19	1036.35	6.75
26.19	83.15	1037.60	1761.75
25.17	85.34	1037.89	652.50
24.30	87.69	1036.86	963.00
24.07	87.65	1027.83	254.25
21.21	95.95	1006.92	339.75
23.49	96.17	1006.57	38.25
21.79	98.59	1009.42	29.25
25.09	88.33	991.65	4.50
25.39	90.43	1009.66	112.50
23.89	94.54	1009.27	735.75
22.51	99.00	1009.80	607.50
22.90	98.00	1009.90	717.75
21.72	99.00	996.29	513.00
23.18	98.97	800.00	195.75
21.24	99.00	1009.21	474.75
21.63	99.00	1008.89	409.50
20.91	99.00	1008.89	1161.00
23.67	97.80	1009.38	0.00
24.53	92.90	1008.66	0.00

- Second strategy:
  - Perform the correlation analysis between each attribute and target attribute
  - Drop the attributes that are less correlated with target attribute
- Example:
  - Predicting `Rain` (target attribute) based on `Temperature`, `Humidity` and `Pressure`
  - `Rain` dependent on `Temperature`, `Humidity` and `Pressure`
  - Correlation analysis of `Temperature`, `Humidity`, `Pressure` with `Rain`

Dimensionality Reduction

Tuple (Data Vector) – Attribute (Dimension)

Temperature	Humidity	Pressure	Rain	Moisture
25.47	82.19	1036.35	6.75	0.00
26.19	83.15	1037.60	1761.75	5.69
25.17	85.34	1037.89	652.50	6.85
24.30	87.69	1036.86	963.00	6.04
24.07	87.65	1027.83	254.25	31.24
21.21	95.95	1006.92	339.75	100.00
23.49	96.17	1006.57	38.25	93.20
21.79	98.59	1009.42	29.25	5.77
25.09	88.33	991.65	4.50	4.29
25.39	90.43	1009.66	112.50	3.62
23.89	94.54	1009.27	735.75	3.76
22.51	99.00	1009.80	607.50	4.03
22.90	98.00	1009.90	717.75	3.83
21.72	99.00	996.29	513.00	3.04
23.18	98.97	800.00	195.75	3.00
21.24	99.00	1009.21	474.75	3.05
21.63	99.00	1008.89	409.50	3.00
20.91	99.00	1008.89	1161.00	3.20
23.67	97.80	1009.38	0.00	2.04
24.53	92.90	1008.66	0.00	1.80

- A tuple (one row) is referred as a vector
- Attribute is referred as dimension
- In this example:
  - Number of vectors = number of rows = 20
  - Dimension of a vector = number of attributes = 5
  - Size of data matrix is 20x5

Tuple (Data Vector)

Dimensionality Reduction

- Data encoding or transformations are applied so as to obtain a reduced or compressed representation of the original data

- If the original data can be reconstructed from compressed data without any loss of information, the data reduction is called lossless
- If only an approximation of the original data can be reconstructed from compressed data, then the data reduction is called lossy
- One of the popular and effective methods of lossy dimensionality reduction is principal component analysis (PCA)

Principal Component Analysis (PCA)

- Suppose data to be reduced consist of N tuples (or data vectors) described by d-attributes (d -dimensions)

$$\mathrm{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathrm{R}^d$$

$$\mathbf{x}_n = [x_{n1} \;\; x_{n2} \;\; \ldots \;\; x_{nd}]^\mathsf{T}$$

- Let $\mathbf{q}_i$, where $i = 1, 2, \ldots, d$ be the d orthonormal vectors in the d –dimensional space, $\mathbf{q}_i \in \mathrm{R}^d$
  - These are unit vectors that each point in a direction perpendicular to the others

$$\mathbf{q}_i^\mathsf{T}\mathbf{q}_j = 0 \quad \forall i \neq j$$

$$\mathbf{q}_i^\mathsf{T}\mathbf{q}_i = 1$$

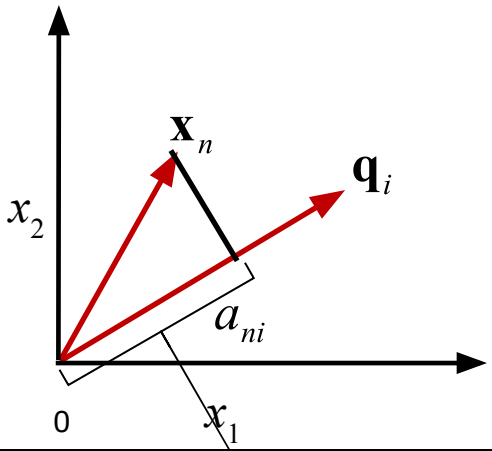
- PCA searches for l orthonormal vectors that can best be used to represent the data, where $l < d$

Principal Component Analysis (PCA)

- These orthonormal vectors are also called as direction of projection
- The original data (each of the tuples (data vectors), $\mathbf{x}_n$) is then projected onto each of the l orthonormal vectors get the principal components

$$a_{ni} = \mathbf{q}_i^{\mathsf{T}} \mathbf{x}_n \quad \forall i = 1, 2, ..., l$$

  - a_{ni} is an i^{th} principal component of $\mathbf{x}_n$
- This transform each of the d – dimensional vectors (i.e. tuples) to l – dimensional vectors



$$\mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \dots \\ x_{nd} \end{bmatrix} \rightarrow \mathbf{a}_n = \begin{bmatrix} a_{n1} \\ a_{n2} \\ \dots \\ a_{nl} \end{bmatrix}$$

- Task:
  - How to obtain the orthonormal vectors?
  - Which l orthonormal vectors to choose?

Principal Component Analysis (PCA)

- Thus the original data is projected onto much smaller space, resulting in dimensionality reduction
- It combines the essence of attributes by creating an alternative, smaller set of variables (attributes)
- It is possible to reconstruct the good approximation of original data, $\mathbf{x}_n$, as linear combination of the direction of projection, $\mathbf{q}_i$, and the principal components, a_{ni}

$$\hat{\mathbf{X}}_n = \sum_{i=1}^{l} a_{ni}\mathbf{q}_i$$

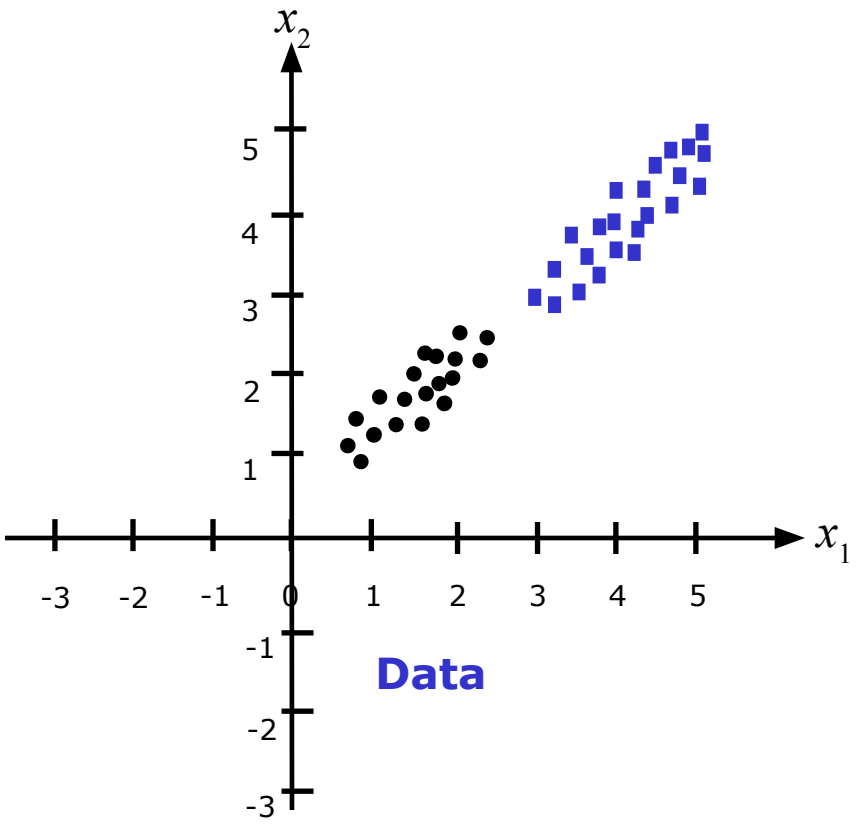
  - $\hat{\mathbf{x}}_n$ is approximation of original tuple $\mathbf{x}_n$

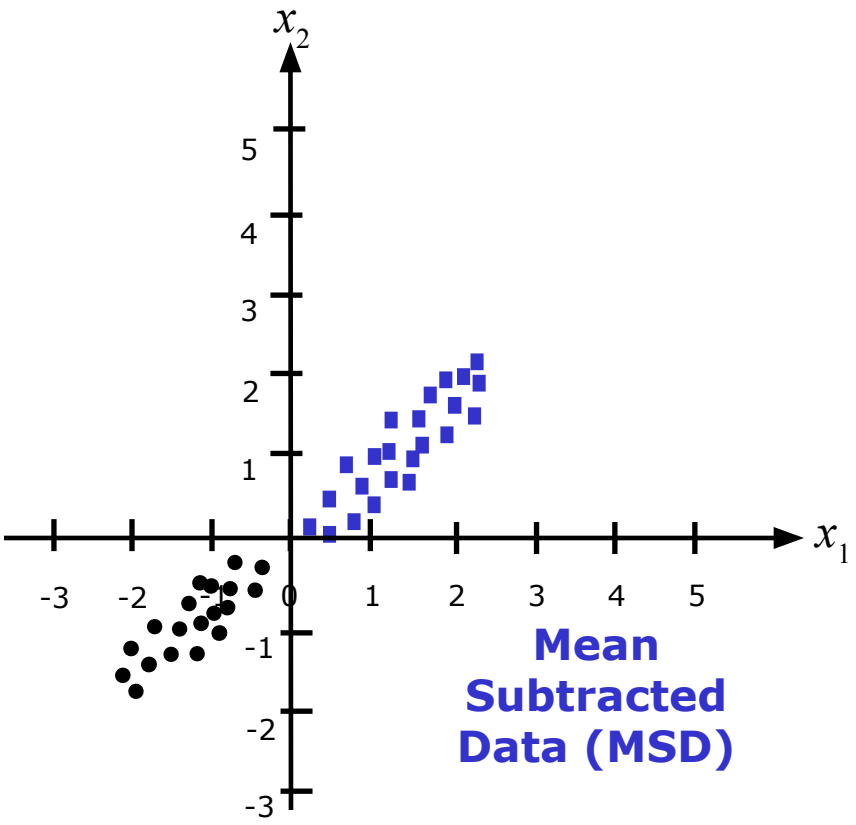
- The Euclidean distance between the original and approximated tuples give the error in reconstruction

$$Error = \|\mathbf{X}_n - \hat{\mathbf{X}}_n\| = \sqrt{\sum_{i=1}^{d}(x_{ni} - \hat{x}_{ni})^2}$$

PCA for Dimension Reduction

- Given: Data with N samples, $\mathrm{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathrm{R}^d$
- Remove mean for each attribute (dimension) in data samples (tuples) to obtain the mean subtracted samples





PCA for Dimension Reduction

- Given: Data with N samples, $\mathrm{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathrm{R}^d$
- Remove mean for each attribute (dimension) in data samples (tuples) to obtain the mean subtracted samples
- Then construct a data matrix $\mathbf{X}$ using the mean subtracted samples, $\mathbf{X} \in \mathrm{R}^{N \times d}$
  - Each row of the matrix $\mathbf{X}$ corresponds to 1 sample (tuple or a data vector)
- Compute a correlation matrix $\mathbf{C}=\mathbf{X}^{\mathsf{T}}\mathbf{X}$

	A_1	A_2		A_d
$\mathbf{x}_1$	x_{11}	x_{12}	…	x_{1d}
$\mathbf{x}_2$	x_{21}	x_{22}	…	x_{2d}
	.	.	.	.
	.	.	.	.
	.	.	.	.
$\mathbf{x}_N$	x_{N1}	x_{N2}	…	x_{Nd}

PCA for Dimension Reduction

- Given: Data with N samples, $\mathrm{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathrm{R}^d$
- Remove mean for each attribute (dimension) in data samples (tuples) to obtain the mean subtracted samples
- Then construct a data matrix $\mathbf{X}$ using the mean subtracted samples, $\mathbf{X} \in \mathrm{R}^{N \times d}$
  - Each row of the matrix $\mathbf{X}$ corresponds to 1 sample (tuple or a data vector)
- Compute a correlation matrix $\mathbf{C}=\mathbf{X}^{\mathsf{T}}\mathbf{X}$

$\mathbf{X}$

$\mathbf{x}_1 - \boldsymbol{\mu}$	$x_{11}-\mu_1$	$x_{12}-\mu_2$	…	$x_{1d}-\mu_d$
$\mathbf{x}_2 - \boldsymbol{\mu}$	$x_{21}-\mu_1$	$x_{22}-\mu_2$	…	$x_{2d}-\mu_d$
	.	.	.	.
	.	.	.	.
	.	.	.	.
$\mathbf{x}_N - \boldsymbol{\mu}_d$	$x_{N1}-\mu_1$	$x_{N2}-\mu_2$	…	$x_{Nd}-\mu_d$

PCA for Dimension Reduction

- Given: Data with N samples, $\mathrm{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathrm{R}^d$
- Remove mean for each attribute (dimension) in data samples (tuples) to obtain the mean subtracted samples
- Then construct a data matrix $\mathbf{X}$ using the mean subtracted samples, $\mathbf{X} \in \mathrm{R}^{N \times d}$
  - Each row of the matrix $\mathbf{X}$ corresponds to 1 sample (tuple or a data vector)
- Compute a correlation matrix $\mathbf{C}=\mathbf{X}^T\mathbf{X}$

$\mathbf{X}^T$

$x_{11}-\mu_1$	$x_{21}-\mu_1$	…	$x_{N1}-\mu_1$
$x_{12}-\mu_2$	$x_{22}-\mu_2$	…	$x_{N2}-\mu_2$
…	…	…	…
$x_{1d}-\mu_d$	$x_{2d}-\mu_d$	…	$x_{Nd}-\mu_d$
$(\mathbf{x}_1-\boldsymbol{\mu})^T$	$(\mathbf{x}_2-\boldsymbol{\mu})^T$		$(\mathbf{x}_N-\boldsymbol{\mu})^T$

$\mathbf{X}$

$x_{11}-\mu_1$	$x_{12}-\mu_2$	…	$x_{1d}-\mu_d$	$\mathbf{x}_1-\boldsymbol{\mu}$
$x_{21}-\mu_1$	$x_{22}-\mu_2$	…	$x_{2d}-\mu_d$	$\mathbf{x}_2-\boldsymbol{\mu}$
.	.	.	.	
.	.	.	.	
.	.	.	.	
$x_{N1}-\mu_1$	$x_{N2}-\mu_2$	…	$x_{Nd}-\mu_d$	$\mathbf{x}_N-\boldsymbol{\mu}_d$

PCA for Dimension Reduction

- Compute a correlation matrix $\mathbf{C}=\mathbf{X}^T\mathbf{X}$

$\mathbf{X}^T$

$x_{11}-\mu_1$	$x_{21}-\mu_1$	…	$x_{N1}-\mu_1$
$x_{12}-\mu_2$	$x_{22}-\mu_2$	…	$x_{N2}-\mu_2$
…	…	…	…
$x_{1d}-\mu_d$	$x_{2d}-\mu_d$	…	$x_{Nd}-\mu_d$

$\mathbf{X}$

$x_{11}-\mu_1$	$x_{12}-\mu_2$	…	$x_{1d}-\mu_d$
$x_{21}-\mu_1$	$x_{22}-\mu_2$	…	$x_{2d}-\mu_d$
.	.	.	.
.	.	.	.
.	.	.	.
$x_{N1}-\mu_1$	$x_{N2}-\mu_2$	…	$x_{Nd}-\mu_d$

$\mathbf{X}^T\mathbf{X}$

$-\hat{x}_1$ is approximation of original tuple x_1	$-\hat{x}_1$ is approximation of original tuple x_1	…	$-\hat{x}_1$ is approximation of original tuple x_1
$-\hat{x}_2$ is approximation of original tuple x_2	$-\hat{x}_2$ is approximation of original tuple x_2	…	$-\hat{x}_2$ is approximation of original tuple x_2
…	…	…	…
$-\hat{x}_n$ is approximation of original tuple x_n	$-\hat{x}_n$ is approximation of original tuple x_n	…	$-\hat{x}_n$ is approximation of original tuple x_n

PCA for Dimension Reduction

- Given: Data with N samples, $\mathrm{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathrm{R}^d$
- Remove mean for each attribute (dimension) in data samples (tuples) to obtain the mean subtracted samples
- Then construct a data matrix $\mathbf{X}$ using the mean subtracted samples, $\mathbf{X} \in \mathrm{R}^{N \times d}$
  - Each row of the matrix $\mathbf{X}$ corresponds to 1 sample (tuple or a data vector)
- Compute a correlation matrix $\mathbf{C}=\mathbf{X}^{\mathsf{T}}\mathbf{X}$

$\mathbf{X}^{\mathsf{T}}\mathbf{X}$

Var(A1)	COV(A1, A2)	…	COV(A1, Ad)
COV(A2, A1)	Var(A2)	…	COV(A2, Ad)
…	…	…	…
COV(Ad, A1)	COV(Ad, A2)	…	Var(Ad)

PCA for Dimension Reduction

- Given: Data with N samples, $\mathrm{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathrm{R}^d$
- Remove mean for each attribute (dimension) in data samples (tuples) to obtain the mean subtracted samples
- Then construct a data matrix $\mathbf{X}$ using the mean subtracted samples, $\mathbf{X} \in \mathrm{R}^{N \times d}$
  - Each row of the matrix $\mathbf{X}$ corresponds to 1 sample (tuple or a data vector)
- Compute a correlation matrix $\mathbf{C}=\mathbf{X}^{\mathsf{T}}\mathbf{X}$
- Perform the eigen analysis of correlation matrix $\mathbf{C}$

$$\mathbf{C}\mathbf{q}_i = \lambda_i \mathbf{q}_i \quad \forall i = 1, 2, ..., d$$

  - As correlation matrix (covariance matrix) is symmetric matrix and positive semidefinite,
    - Each eigenvalues λ_i are distinct and non-negative.
    - Eigenvectors $\mathbf{q}_i$ corresponding to each eigenvalues are orthonormal vectors
    - Eigenvalues indicate the strength of eigenvectors or variance of projected data in the direction of eigenvector

PCA for Dimension Reduction

- Project the $\mathbf{x}_n$ onto each of the directions (eigenvectors) to get the principal components

$$a_{ni} = \mathbf{q}_i^{\mathsf{T}} \mathbf{x}_n \quad \forall i = 1, 2, ..., d$$

  - a_{ni} is an i^{th} principal component of $\mathbf{x}_n$

- Thus, each training example $\mathbf{x}_n$ is transformed to a new representation $\mathbf{a}_n$ by projecting on to d-orthonormal basis (eigenvectors)

$$\mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \dots \\ x_{nd} \end{bmatrix} \rightarrow \mathbf{a}_n = \begin{bmatrix} a_{n1} \\ a_{n2} \\ \dots \\ a_{nd} \end{bmatrix}$$

  - It is possible to reconstruct the original data, $\mathbf{x}_n$, without error as linear combination of the direction of projection, $\mathbf{q}_i$, and the principal components, a_{ni}

$$\mathbf{x}_n = \sum_{i=1}^{d} a_{ni} \mathbf{q}_i$$

PCA for Dimension Reduction

- In general, we are interested in representing the data using fewer dimensions such that the data has high variance along these dimensions
- Idea: Select l out of d orthonormal basis vectors (eigenvectors) that contain high variance of data (i.e. more information content)
- Rank order the eigenvalues (λ_i's) such that

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$$

- Based on the Definition 1, consider the l (l << d) eigenvectors corresponding to l significant eigenvalues
  - Definition 1: Let $\lambda_1, \lambda_2, \ldots, \lambda_d$, be the eigenvalues of an d x d matrix $\mathbf{A}$. λ_1 is called the dominant (significant) eigenvalue of $\mathbf{A}$ if $|\lambda_1| \geq |\lambda_i|, \; i = 1, 2, \ldots, d$

PCA for Dimension Reduction

- Project the $\mathbf{x}_n$ onto each of the l directions (eigenvectors) to get reduced dimensional representation

$$a_{ni} = \mathbf{q}_i^{\mathsf{T}} \mathbf{x}_n \quad \forall i = 1, 2, ..., l$$

- Thus, each training example $\mathbf{x}_n$ is transformed to a new reduced dimensional representation $\mathbf{a}_n$ by projecting on to l-orthonormal basis vectors (eigenvectors)

$$\mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \dots \\ x_{nd} \end{bmatrix} \rightarrow \mathbf{a}_n = \begin{bmatrix} a_{n1} \\ a_{n2} \\ \dots \\ a_{nl} \end{bmatrix}$$

- The eigenvalue λ_i correspond to the variance of projected data

PCA for Dimension Reduction

- Since the strongest l directions are considered for obtaining reduced dimensional representation, it should be possible to reconstruct a good approximation of the original data
- An approximation of original data, $\mathbf{x}_n$, is obtained as linear combination of the direction of projection (strongest eigenvectors), $\mathbf{q}_i$, and the principal components, a_i

$$\hat{\mathbf{X}}_n = \sum_{i=1}^{l} a_i \mathbf{q}_i$$

  - $\hat{\mathbf{x}}_n$ is approximation of original tuple $\mathbf{x}_n$

Dimensionality Reduction
Principal Component Analysis (PCA)

Dimensionality Reduction

- Data encoding or transformations are applied so as to obtain a reduced or compressed representation of the original data

- If the original data can be reconstructed from compressed data without any loss of information, the data reduction is called lossless
- If only an approximation of the original data can be reconstructed from compressed data, then the data reduction is called lossy
- One of the popular and effective methods of lossy dimensionality reduction is principal component analysis (PCA)

Principal Component Analysis (PCA)

- Suppose data to be reduced consist of N tuples (or data vectors) described by d-attributes (d -dimensions)

$$\mathrm{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathrm{R}^d$$

$$\mathbf{x}_n = [x_{n1} \;\; x_{n2} \;\; \ldots \;\; x_{nd}]^\mathsf{T}$$

- Let $\mathbf{q}_i$, where $i = 1, 2, \ldots, d$ be the d orthonormal vectors in the d –dimensional space, $\mathbf{q}_i \in \mathrm{R}^d$
  - These are unit vectors that each point in a direction perpendicular to the others

$$\mathbf{q}_i^\mathsf{T}\mathbf{q}_j = 0 \quad \forall i \neq j$$

$$\mathbf{q}_i^\mathsf{T}\mathbf{q}_i = 1$$

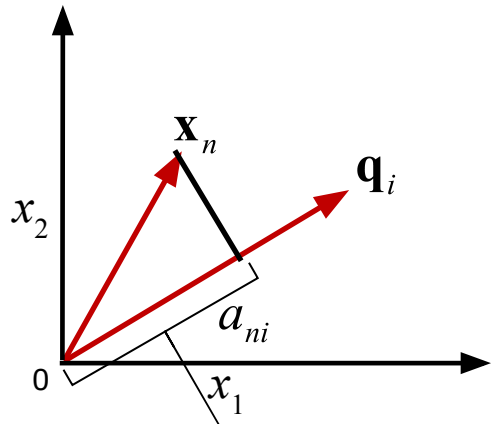
- These orthonormal vectors are also called as direction of projection

Principal Component Analysis (PCA)

- PCA searches for l orthonormal vectors that can best be used to represent the data, where $l < d$
- The original data (each of the tuples (data vectors), $\mathbf{x}_n$) is then projected onto each of the l orthonormal vectors get the principal components

$$a_{ni} = \mathbf{q}_i^{\mathsf{T}} \mathbf{x}_n \quad \forall i = 1, 2, ..., l$$

  - a_{ni} is an i^{th} principal component of $\mathbf{x}_n$
- This transform each of the d – dimensional vectors (i.e. tuples) to l – dimensional vectors



$$\mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \cdots \\ x_{nd} \end{bmatrix} \rightarrow \mathbf{a}_n = \begin{bmatrix} a_{n1} \\ a_{n2} \\ \cdots \\ a_{nl} \end{bmatrix}$$

- Task:
  - How to obtain the orthonormal vectors?
  - Which l orthonormal vectors to choose from d orthonormal vectors?

PCA: Basic Procedure

- Given: Data with N samples, $\mathrm{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathrm{R}^d$

1. Remove mean for each attribute (dimension) in data samples (tuples)
2. Then construct a data matrix $\mathbf{X}$ using the mean subtracted samples, $\mathbf{X} \in \mathrm{R}^{N \times d}$
   - Each row of the matrix $\mathbf{X}$ corresponds to 1 sample (tuple)
3. Compute a correlation matrix $\mathbf{C}=\mathbf{X}^{\mathsf{T}}\mathbf{X}$
   - This correlation matrix is equivalent to covariance matrix from original data matrix:

$$\mathbf{C}\mathbf{q}_i = \lambda_i \mathbf{q}_i \quad \forall i = 1, 2, ..., d$$

4. Perform the eigen analysis of covariance matrix $\mathbf{C}$
   1. As covariance matrix is symmetric matrix,
      1. Each eigenvalues λ_i are distinct and non-negative
      - Eigenvectors $\mathbf{q}_i$ corresponding to each eigenvalues are orthonormal vectors
      1. Eigenvalues indicate the strength of eigenvectors or variance of projected data in the direction of eigenvector

PCA for Dimension Reduction

- In general, we are interested in representing the data using fewer dimensions such that the data has high variance along these dimensions

5. Rank order the eigenvalues (λ_i's) (descending order of λ_i's) such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$

6. Consider the l ($l << d$) eigenvectors corresponding to l significant eigenvalues

7. Project the $\mathbf{x}_n$ onto each of the l directions (eigenvectors) to get reduced dimensional representation in terms of principal components

$$a_{ni} = \mathbf{q}_i^T \mathbf{x}_n \quad \forall i = 1, 2, \ldots, l$$

a_{ni} is an i^{th} principal component of $\mathbf{x}_n$

PCA for Dimension Reduction

8. Thus, each training example $\mathbf{x}_n$ is transformed to a new reduced dimensional representation $\mathbf{a}_n$ by projecting on to l-orthonormal basis

$$\mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \dots \\ x_{nd} \end{bmatrix} \rightarrow \mathbf{a}_n = \begin{bmatrix} a_{n1} \\ a_{n2} \\ \dots \\ a_{nl} \end{bmatrix}$$

- Observations:
  - The new reduced representation $\mathbf{a}_n$ is uncorrelated
  - The eigenvalue λ_i correspond to the variance of projected data (reduced representation)
- Note:
  - The number l is chosen experimentally (empirically) by observing the values of eigenvalue
  - If the data is projected onto all the eigenvectors, $\mathbf{x}_n$ is transformed to a new representation $\mathbf{a}_n$ with d-dimension

    The new representation is uncorrelated

Illustration: PCA

Temperature	Humidity	Pressure	Rain	Moisture
25.47	82.19	1036.35	6.75	0.00
26.19	83.15	1037.60	1761.75	5.69
25.17	85.34	1037.89	652.50	6.85
24.30	87.69	1036.86	963.00	6.04
24.07	87.65	1027.83	254.25	31.24
21.21	95.95	1006.92	339.75	100.00
23.49	96.17	1006.57	38.25	93.20
21.79	98.59	1009.42	29.25	5.77
25.09	88.33	991.65	4.50	4.29
25.39	90.43	1009.66	112.50	3.62
23.89	94.54	1009.27	735.75	3.76
22.51	99.00	1009.80	607.50	4.03
22.90	98.00	1009.90	717.75	3.83
21.72	99.00	996.29	513.00	3.04
23.18	98.97	800.00	195.75	3.00
21.24	99.00	1009.21	474.75	3.05
21.63	99.00	1008.89	409.50	3.00
20.91	99.00	1008.89	1161.00	3.20
23.67	97.80	1009.38	0.00	2.04
24.53	92.90	1008.66	0.00	1.80

- Atmospheric Data:
  - N = Number tiples (data vectors) = 20
  - d = Number of attributes (dimension) = 5
- Mean of each dimension:

23.42	93.64	1003.55	448.88	14.4

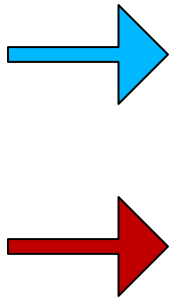


Illustration: PCA

Temperature	Humidity	Pressure	Rain	Moisture
2.05	-11.45	32.80	-442.13	-14.37
2.77	-10.49	34.05	1312.88	-8.68
1.75	-8.30	34.34	203.63	-7.52
0.88	-5.95	33.31	514.13	-8.33
0.65	-5.99	24.28	-194.63	16.87
-2.21	2.32	3.37	-109.13	85.63
0.07	2.54	3.02	-410.63	78.83
-1.63	4.96	5.87	-419.63	-8.60
1.67	-5.31	-11.90	-444.38	-10.08
1.97	-3.21	6.11	-336.38	-10.75
0.47	0.91	5.72	286.88	-10.61
-0.91	5.36	6.25	158.63	-10.34
-0.52	4.36	6.35	268.88	-10.54
-1.70	5.36	-7.26	64.13	-11.33
-0.24	5.33	-203.55	-253.13	-11.37
-2.18	5.36	5.66	25.88	-11.32
-1.79	5.36	5.34	-39.38	-11.37
-2.51	5.36	5.34	712.13	-11.17
0.25	4.16	5.83	-448.88	-12.33
1.11	-0.73	5.11	-448.88	-12.57

- Step1: Subtract mean from each attribute

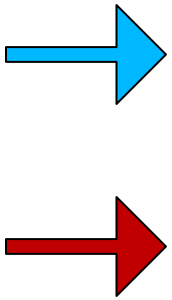


Illustration: PCA

- Step2: Multiply the mean subtracted data matrix with its transpose (i.e., covariance matrix of original data matrix)

2.64	-8.21	14.13	16.98	-9.66
-8.21	35.05	-117.00	-459.95	13.34
14.13	-117.00	2478.61	5420.36	80.09
16.98	-459.95	5420.36	215276.95	-2427.97
-9.66	13.34	80.09	-2427.97	832.14

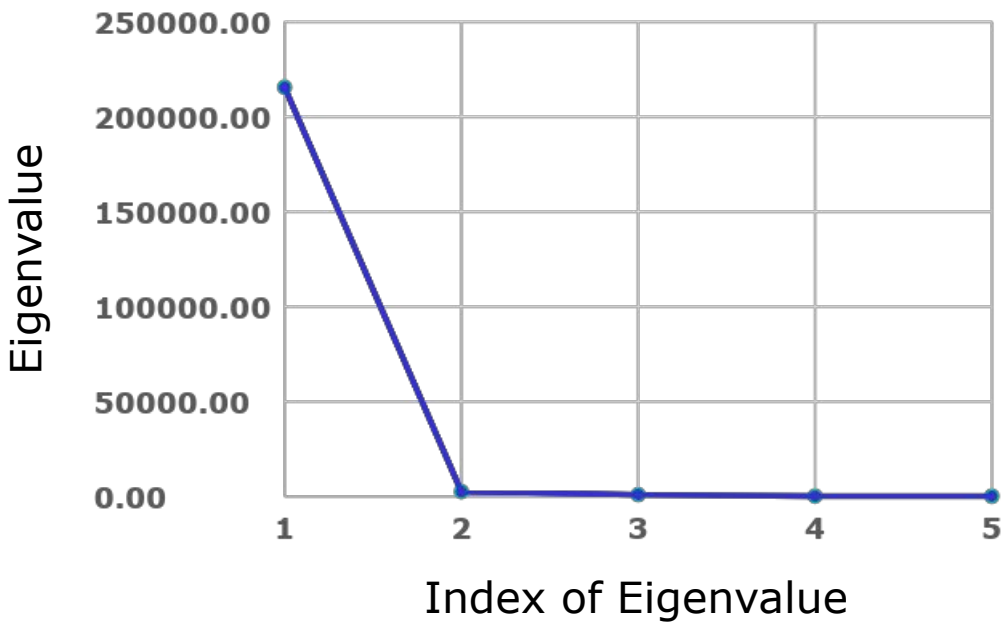
Illustration: PCA

Eigen Values

215443.33	2358.36	792.30	30.88	0.52

Eigen Vectors

-0.0001	0.0056	-0.0137	0.2498	0.9682
0.0021	-0.0448	0.0232	-0.9669	0.2501
-0.0254	0.9946	-0.0892	-0.0469	0.0051
-0.9996	-0.0244	0.0136	-0.0007	0.0004
0.0113	0.0906	0.9956	0.0218	0.0080



- Step3: Perform Eigen analysis on correlation matrix
  - Get eigenvalues and eigenvectors
- Step4: Sort the eigenvalues in descending order
- Step5: Arrange the eigenvectors in the descending order of their corresponding eigenvalues
- Step6: Consider the two leading (significant) eigenvalues and their corresponding eigenvectors

Illustration: PCA

a_1	a_2
440.93	42.62
-1313.35	1.55
-204.52	28.89
-514.88	20.11
194.11	30.69
109.97	13.65
411.28	20.04
419.23	15.05
444.38	-1.66
335.96	13.46
-287.03	-2.31
-158.83	1.16
-269.04	-1.40
-64.03	-10.06
258.09	-197.54
-26.13	3.72
39.11	4.99
-712.10	-13.32
448.42	15.44
448.43	14.93

- Step7: Project the mean subtracted data matrix onto the selected two eigenvectors corresponding to leading eigenvalues

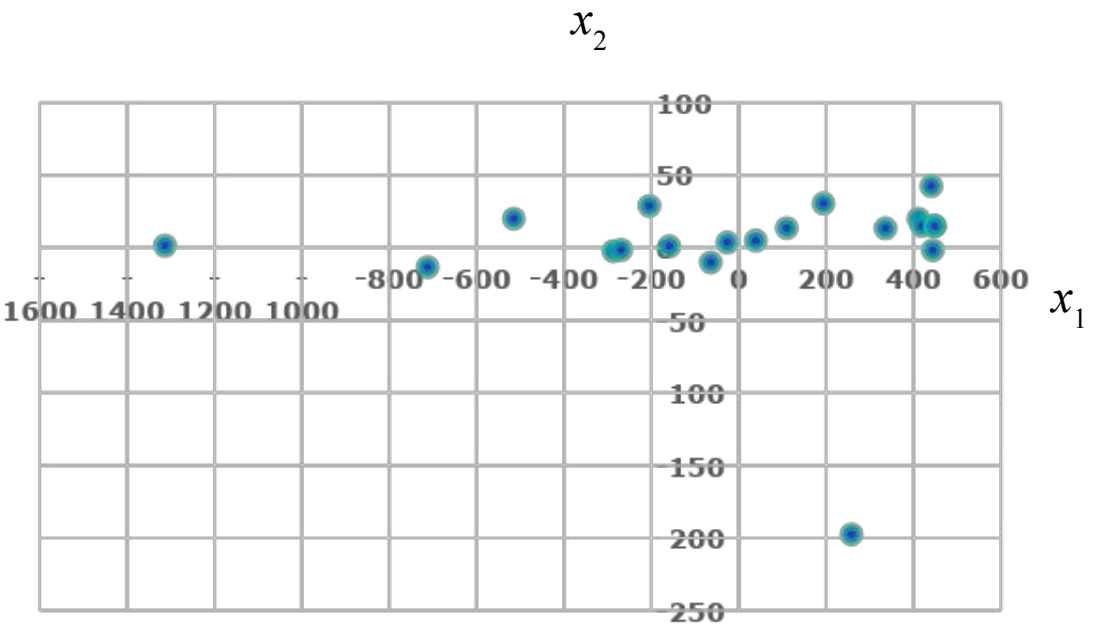


Illustration: PCA

a_1	a_2
440.93	42.62
-1313.35	1.55
-204.52	28.89
-514.88	20.11
194.11	30.69
109.97	13.65
411.28	20.04
419.23	15.05
444.38	-1.66
335.96	13.46
-287.03	-2.31
-158.83	1.16
-269.04	-1.40
-64.03	-10.06
258.09	-197.54
-26.13	3.72
39.11	4.99
-712.10	-13.32
448.42	15.44
448.43	14.93

- Step7: Project the mean subtracted data matrix onto the selected two eigenvectors corresponding to leading eigenvalues

- Covariance matrix of 2-dimensional representation obtained using PCA:

215443.33	0.00
0.00	2358.36

- The reduced representation is uncorrelated

Illustration: PCA – Reconstruction of Data

x_1	x_2	x_3	x_4	x_5	*Error*
0.20	-0.96	31.17	-441.80	8.84	25.59
0.12	-2.89	34.96	1312.80	-14.70	10.09
0.18	-1.73	33.93	203.74	0.30	10.34
0.15	-2.01	33.09	514.19	-4.00	5.91
0.16	-0.96	25.59	-194.78	4.97	12.99
0.07	-0.37	10.78	-110.26	2.48	83.56
0.08	-0.01	9.47	-411.61	6.46	72.70
0.05	0.23	4.31	-419.43	6.10	15.61
-0.05	1.03	-12.96	-444.17	4.87	16.37
0.05	0.12	4.84	-336.16	5.01	16.28
0.01	-0.51	5.01	286.97	-3.45	7.34
0.02	-0.39	5.20	158.74	-1.69	10.49
0.01	-0.52	5.45	268.97	-3.17	8.90
-0.05	0.31	-8.38	64.25	-1.63	11.11
-1.12	9.40	-203.04	-253.17	-14.97	5.52
0.02	-0.22	4.36	26.02	0.04	12.92
0.02	-0.14	3.97	-39.21	0.89	13.64
-0.02	-0.93	4.87	712.14	-9.25	7.06
0.05	0.27	3.95	-448.62	6.47	19.30
0.05	0.30	3.44	-448.62	6.42	19.12

- An approximation of mean subtracted data, $\mathbf{x}_n$, is obtained as linear combination of the direction of projection (strongest eigenvectors), $\mathbf{q}_i$, and the principal components, a_{ni}

$$\hat{\mathbf{x}}_n = \sum_{i=1}^{l} a_{ni}\mathbf{q}_i$$

- Error in reconstruction: The Euclidean distance between the original and approximated tuples

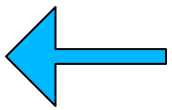


Illustration: PCA - Extended

a_1	a_2	a_3	a_4	a_5
440.93	42.62	-23.53	10.02	-1.01
-1313.35	1.55	5.86	8.18	0.72
-204.52	28.89	-8.00	6.55	-0.18
-514.88	20.11	-4.44	3.89	-0.32
194.11	30.69	11.84	5.31	-0.69
109.97	13.65	83.55	-1.01	-0.91
411.28	20.04	72.69	-0.59	1.17
419.23	15.05	-14.65	-5.38	-0.55
444.38	-1.66	-15.16	6.18	-0.03
335.96	13.46	-15.92	3.29	0.92
-287.03	-2.31	-7.16	-1.44	0.75
-158.83	1.16	-8.56	-6.04	0.48
-269.04	-1.40	-7.30	-5.05	0.66
-64.03	-10.06	-9.61	-5.56	-0.40
258.09	-197.54	3.52	4.25	-0.13
-26.13	3.72	-11.27	-6.26	-0.81
39.11	4.99	-12.18	-6.11	-0.47
-712.10	-13.32	-1.78	-6.78	-0.85
448.42	15.44	-18.80	-4.21	1.03
448.43	14.93	-19.10	0.77	0.63

- Step6: Consider the all the eigenvalues and their corresponding eigenvectors
- Step7: Project the mean subtracted data matrix onto all the eigenvectors
- The resultant 5-dimensional representation is a new transformed representation
- Covariance matrix of new representation obtained using PCA:

215443.33	0.00	0.00	0.00	0.00
0.00	2358.36	0.00	0.00	0.00
0.00	0.00	792.30	0.00	0.00
0.00	0.00	0.00	30.88	0.00
0.00	0.00	0.00	0.00	0.52

- The encoded representation is uncorrelated

Illustration: PCA – Reconstruction of Data

x_1	x_2	x_3	x_4	x_5	*Error*
2.05	-11.45	32.80	-442.13	-14.37	1.79E-14
2.77	-10.49	34.05	1312.88	-8.68	3.83E-14
1.75	-8.29	34.34	203.63	-7.52	2.18E-14
0.88	-5.95	33.31	514.13	-8.33	1.15E-13
0.65	-5.99	24.28	-194.63	16.87	3.43E-14
-2.21	2.31	3.37	-109.13	85.63	1.55E-14
0.07	2.54	3.02	-410.63	78.83	5.73E-14
-1.62	4.96	5.86	-419.63	-8.60	5.87E-14
1.68	-5.31	-11.90	-444.38	-10.08	5.83E-14
1.98	-3.20	6.11	-336.38	-10.76	1.19E-14
0.47	0.90	5.72	286.88	-10.61	5.73E-14
-0.91	5.37	6.24	158.63	-10.34	4.07E-15
-0.51	4.37	6.34	268.88	-10.54	5.73E-14
-1.69	5.37	-7.26	64.13	-11.33	1.52E-14
-0.24	5.34	-203.55	-253.13	-11.37	1.17E-13
-2.18	5.37	5.65	25.88	-11.32	2.66E-15
-1.79	5.37	5.34	-39.38	-11.37	3.23E-15
-2.51	5.37	5.34	712.13	-11.18	1.15E-13
0.25	4.17	5.83	-448.88	-12.34	9.06E-15
1.11	-0.73	5.11	-448.88	-12.57	5.82E-14

- An approximation of mean subtracted data, $\mathbf{x}_n$, is obtained as linear combination of the direction of projection (strongest eigenvectors), $\mathbf{q}_i$, and the principal components, a_{ni}

$$\hat{\mathbf{x}}_n = \sum_{i=1}^{d} a_{ni}\mathbf{q}_i$$

- Error in reconstruction: The Euclidean distance between the original and approximated tuples

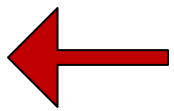


Illustration: PCA – *Projecting Original Data*

a_1	a_2
-32.94	1027.02
-1787.22	985.95
-678.39	1013.28
-988.75	1004.50
-279.76	1015.09
-363.90	998.05
-62.58	1004.44
-54.64	999.45
-29.49	982.74
-137.91	997.85
-760.89	982.09
-632.69	985.56
-742.91	983.00
-537.90	974.34
-215.78	786.85
-499.99	988.11
-434.76	989.39
-1185.97	971.08
-25.45	999.84
-25.44	999.32

- Step6: Consider the two leading (significant) eigenvalues and their corresponding eigenvectors
- Step7: Project the original data matrix *(not mean subtracted)* onto the selected two eigenvectors corresponding to leading eigenvalues
- Covariance matrix of 2-dimensional representation obtained using PCA:

215443.33	0.00
0.00	2358.36

- The reduced representation is uncorrelated

Illustration: PCA – *Projecting Original Data* - Reconstruction of Data

x_1	x_2	x_3	x_4	x_5	*Error*
5.75	-46.08	1022.31	7.87	92.68	160.09
5.70	-47.92	1026.02	1762.45	69.13	147.51
5.74	-46.82	1025.04	653.39	84.14	154.87
5.72	-47.08	1024.19	963.84	79.83	155.29
5.71	-46.06	1016.71	254.88	88.81	147.15
5.63	-45.48	1001.90	339.40	86.31	143.03
5.63	-45.13	1000.61	38.05	90.30	142.58
5.60	-44.89	995.44	30.23	89.93	167.71
5.51	-44.09	978.18	5.50	88.70	158.83
5.60	-44.99	995.96	113.51	88.85	161.81
5.58	-45.60	996.11	736.62	80.38	161.30
5.58	-45.48	996.31	608.39	82.14	165.67
5.58	-45.60	996.56	718.63	80.66	164.32
5.51	-44.78	982.74	513.91	82.20	165.49
4.43	-35.70	788.08	196.49	68.85	151.55
5.58	-45.32	995.47	475.68	83.87	166.72
5.58	-45.24	995.09	410.44	84.73	167.13
5.56	-45.99	995.96	1161.80	74.58	162.85
5.60	-44.85	995.09	1.04	90.30	169.32
5.60	-44.82	994.57	1.05	90.25	165.37

- An approximation of original data, $\mathbf{x}_n$, is obtained as linear combination of the direction of projection (strongest eigenvectors), $\mathbf{q}_i$, and the principal components, a_{ni}

$$\hat{\mathbf{x}}_n = \sum_{i=1}^{l} a_{ni}\mathbf{q}_i$$

- Error in reconstruction: The Euclidean distance between the original and approximated tuples

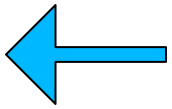


Illustration: PCA – Extended - *Projecting Original Data*

a_1	a_2	a_3	a_4	a_5
-32.94	1027.02	-90.78	-121.73	50.52
-1787.22	985.95	-61.39	-123.57	52.25
-678.39	1013.28	-75.26	-125.20	51.35
-988.75	1004.50	-71.69	-127.86	51.21
-279.76	1015.09	-55.42	-126.44	50.84
-363.90	998.05	16.30	-132.76	50.62
-62.58	1004.44	5.44	-132.34	52.70
-54.64	999.45	-81.90	-137.13	50.98
-29.49	982.74	-82.41	-125.57	51.49
-137.91	997.85	-83.17	-128.46	52.44
-760.89	982.09	-74.41	-133.19	52.28
-632.69	985.56	-75.82	-137.79	52.01
-742.91	983.00	-74.55	-136.80	52.18
-537.90	974.34	-76.87	-137.31	51.13
-215.78	786.85	-63.73	-127.50	51.40
-499.99	988.11	-78.52	-138.01	50.71
-434.76	989.39	-79.44	-137.86	51.06
-1185.97	971.08	-69.03	-138.53	50.67
-25.45	999.84	-86.05	-135.96	52.55
-25.44	999.32	-86.35	-130.98	52.16

- Step6: Consider the all the eigenvalues and their corresponding eigenvectors
- Step7: Project the original data matrix *(not mean subtracted)* onto all the eigenvectors
- The resultant 5-dimensional representation is a new transformed representation
- Covariance matrix of new representation obtained using PCA:

215443.33	0.00	0.00	0.00	0.00
0.00	2358.36	0.00	0.00	0.00
0.00	0.00	792.30	0.00	0.00
0.00	0.00	0.00	30.88	0.00
0.00	0.00	0.00	0.00	0.52

- The encoded representation is uncorrelated

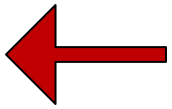
Illustration: PCA – *Projecting Original Data* - Reconstruction of Data

x_1	x_2	x_3	x_4	x_5	*Error*
25.50	82.15	1036.38	6.74	0.05	0.08
26.26	83.20	1037.56	1761.72	5.74	0.11
25.22	85.33	1037.89	652.48	6.89	0.07
24.35	87.69	1036.84	962.98	6.08	0.07
24.11	87.62	1027.85	254.24	31.28	0.07
21.25	95.93	1006.93	339.74	100.05	0.07
23.52	96.14	1006.60	38.23	93.25	0.07
21.83	98.55	1009.44	29.23	5.81	0.07
25.12	88.29	991.69	4.49	4.33	0.08
25.42	90.40	1009.68	112.49	3.66	0.06
23.94	94.53	1009.26	735.72	3.81	0.08
22.56	99.00	1009.80	607.48	4.07	0.07
22.95	97.99	1009.89	717.73	3.88	0.07
21.77	98.99	996.30	512.98	3.08	0.07
23.22	98.95	800.01	195.74	3.03	0.06
21.28	98.99	1009.21	474.73	3.10	0.07
21.67	98.99	1008.90	409.48	3.04	0.06
20.96	99.02	1008.87	1160.98	3.24	0.07
23.70	97.76	1009.41	-0.01	2.08	0.07
24.56	92.86	1008.68	-0.02	1.84	0.07

- An approximation of original data, $\mathbf{x}_n$, is obtained as linear combination of the direction of projection (strongest eigenvectors), $\mathbf{q}_i$, and the principal components, a_{ni}

$$\hat{\mathbf{x}}_n = \sum_{i=1}^{l} a_{ni}\mathbf{q}_i$$

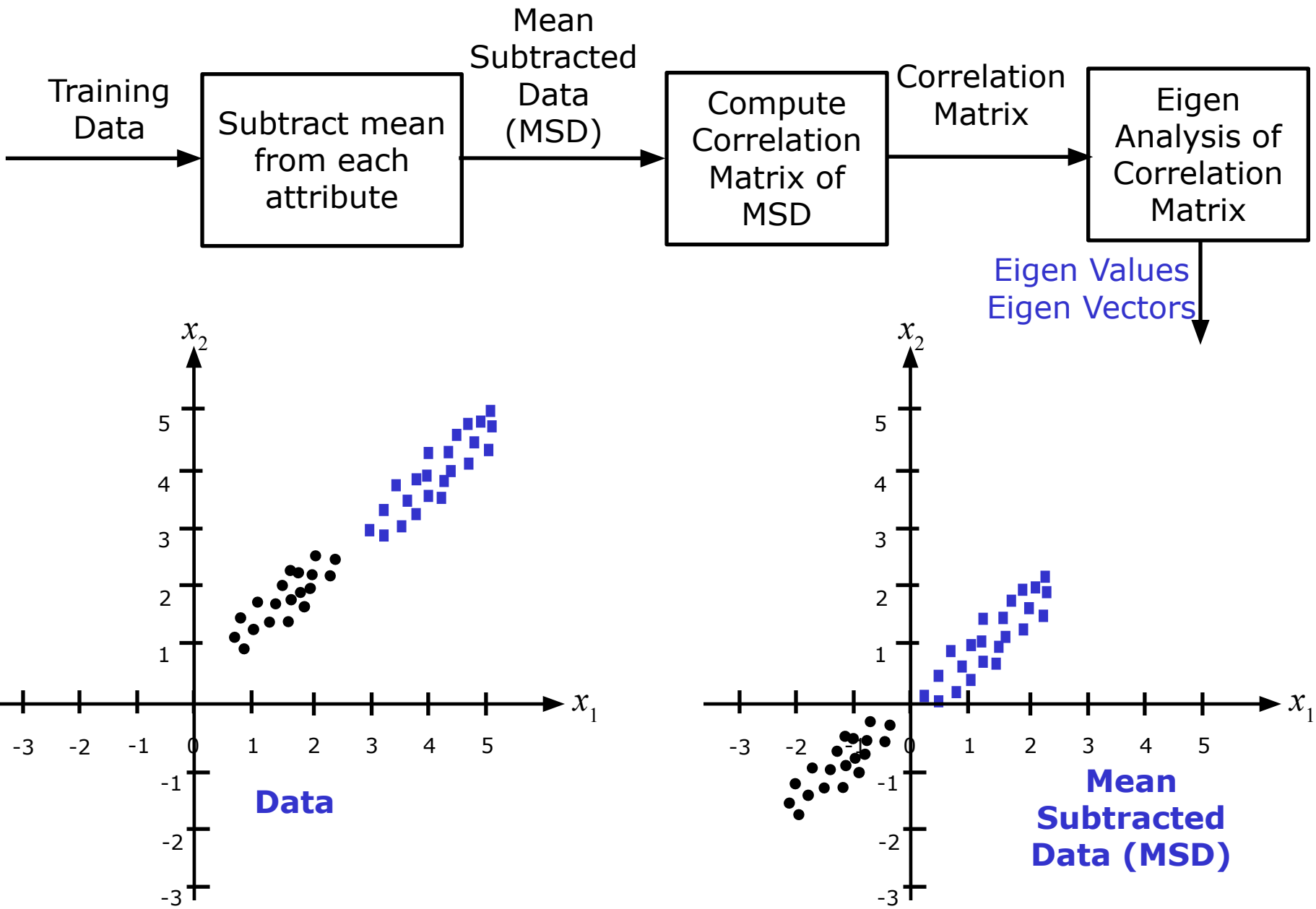
- Error in reconstruction: The Euclidean distance between the original and approximated tuples



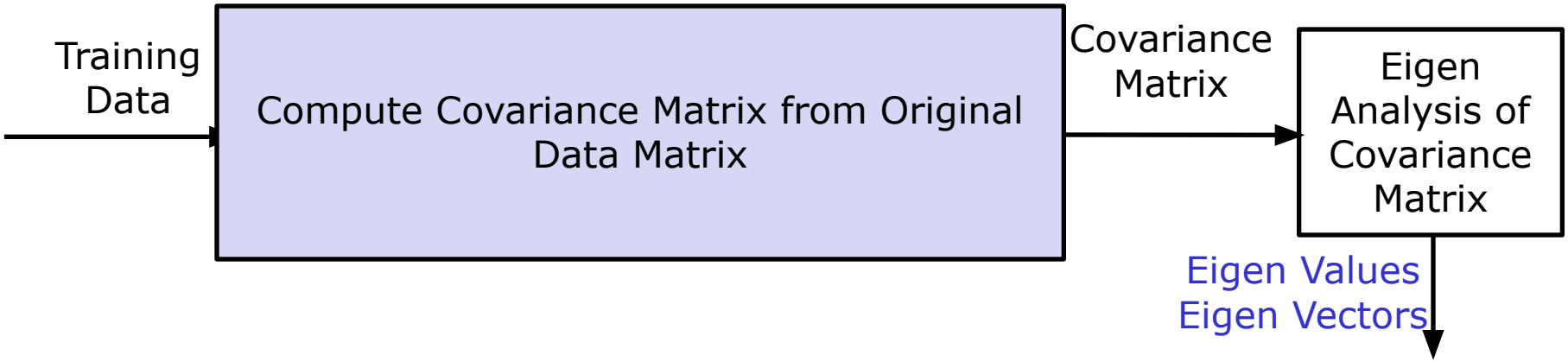
PCA during Model Building

- Model building and prediction using machine learning involve two stages:
  - Training stage: Model building
  - Test stage: Prediction using the built model
- Training stage: Perform the PCA on training data
  - Obtain the l direction of projection (eigenvectors) corresponding to l significant eigenvalues
  - Obtain the reduced dimension representation of training data by projecting training data on to l eigenvectors
- Test stage:
  - Obtain the reduced dimension representation of test data (test data vector(s)) by projecting it on to l eigenvectors obtained during training phase

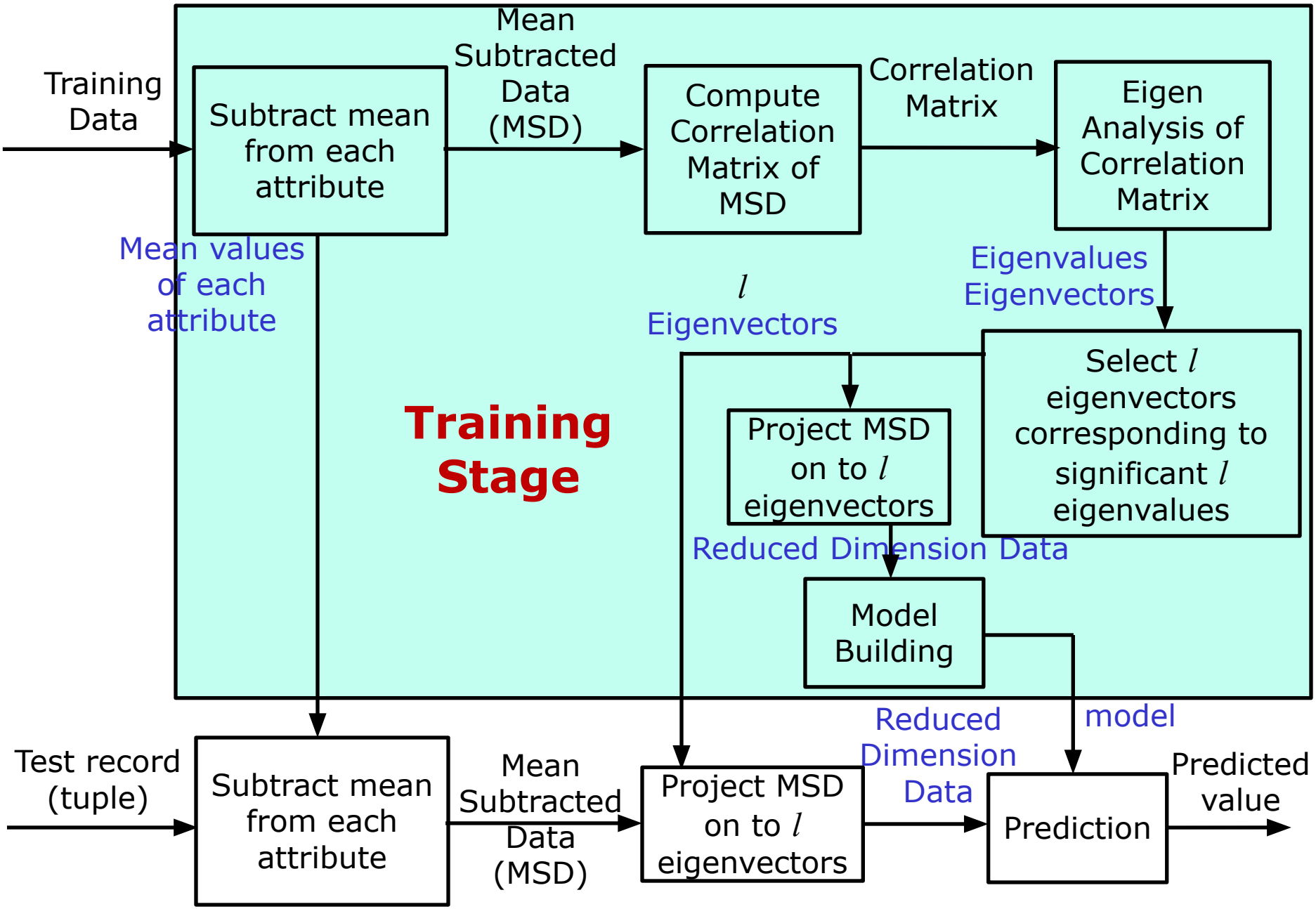
PCA during Model Building



PCA during Model Building



PCA during Model Building



Summary: Dimensionality Reduction

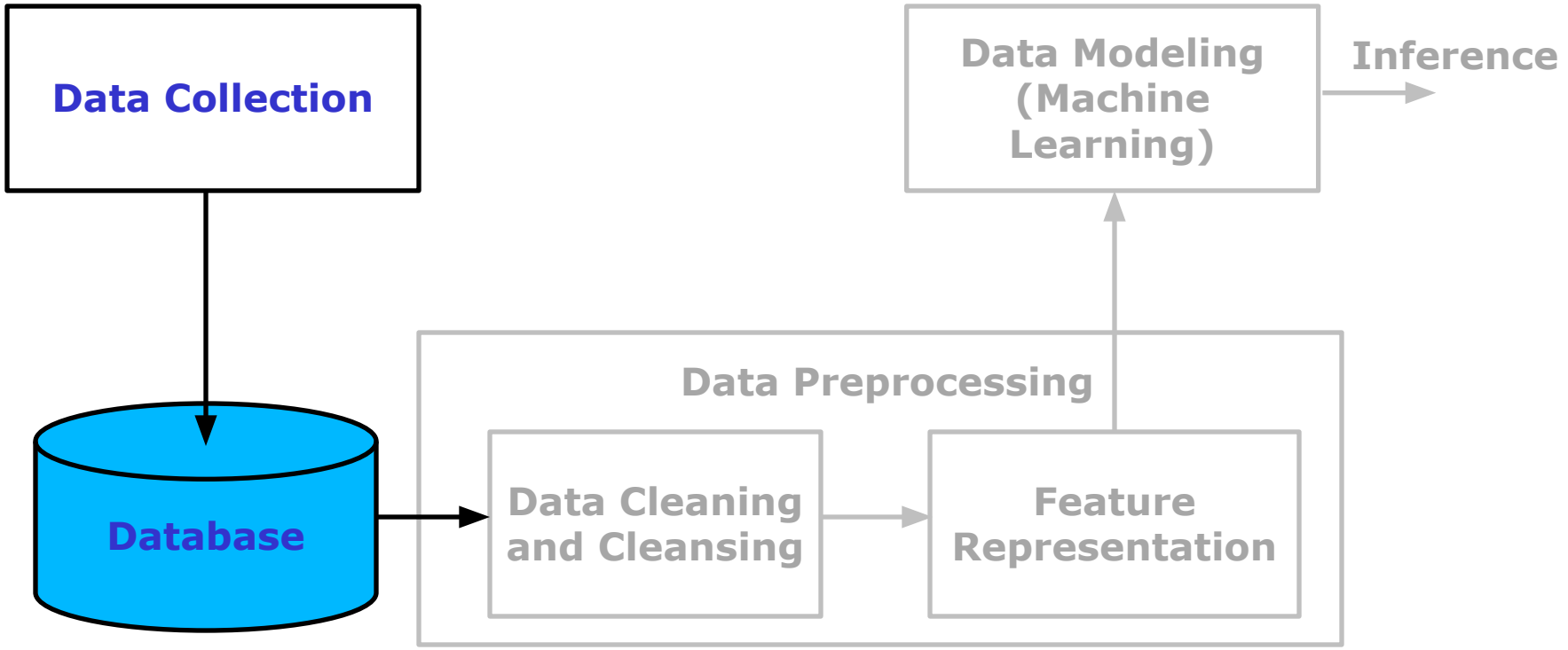
- This technique encodes (transforms) the original representation of data into a reduced or compressed representation of the original data
- Principal component analysis (PCA) is one of the popular and effective methods of lossy dimensionality reduction
- PCA can be used to obtain
  - uncorrelated reduced dimensional representation of data

    OR
  - uncorrelated transformed (encoded) representation with the number of dimension same as original data

Introduction to Machine Learning: Data Modeling

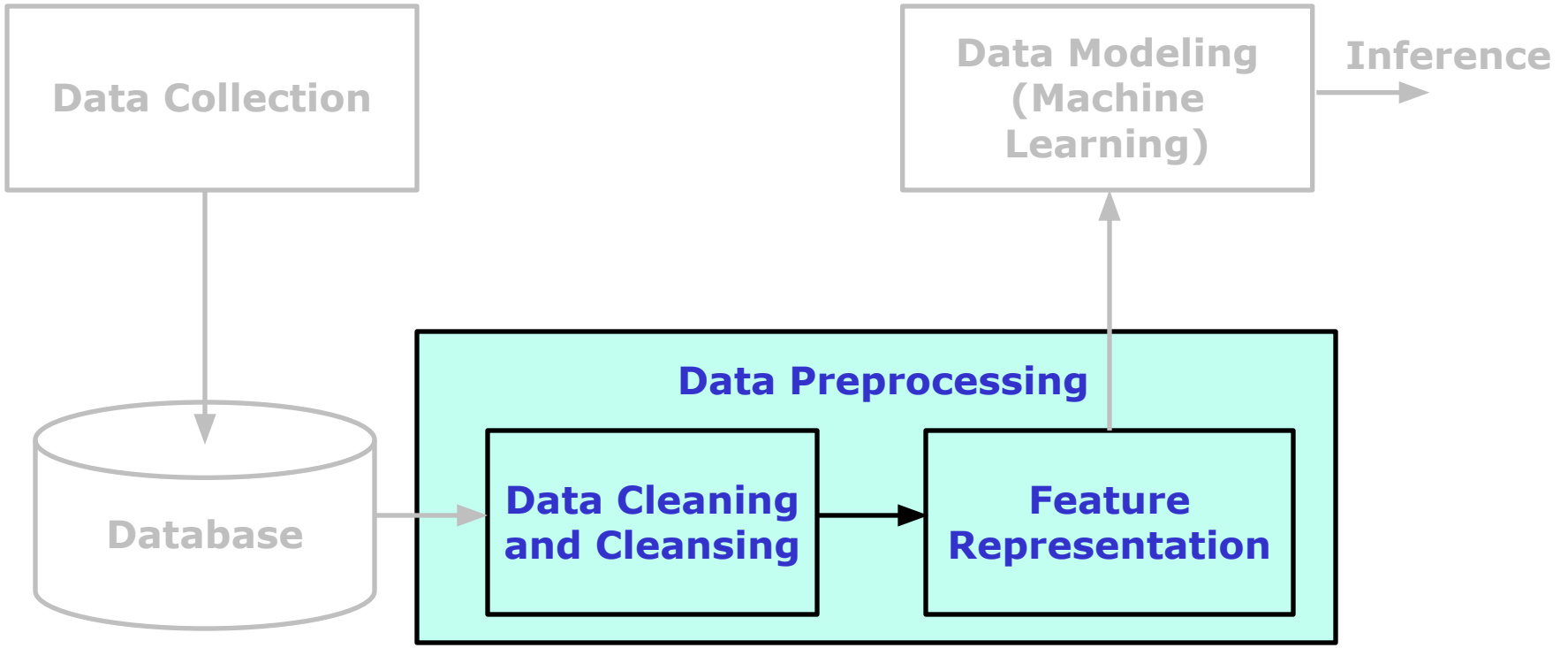
Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data



Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data

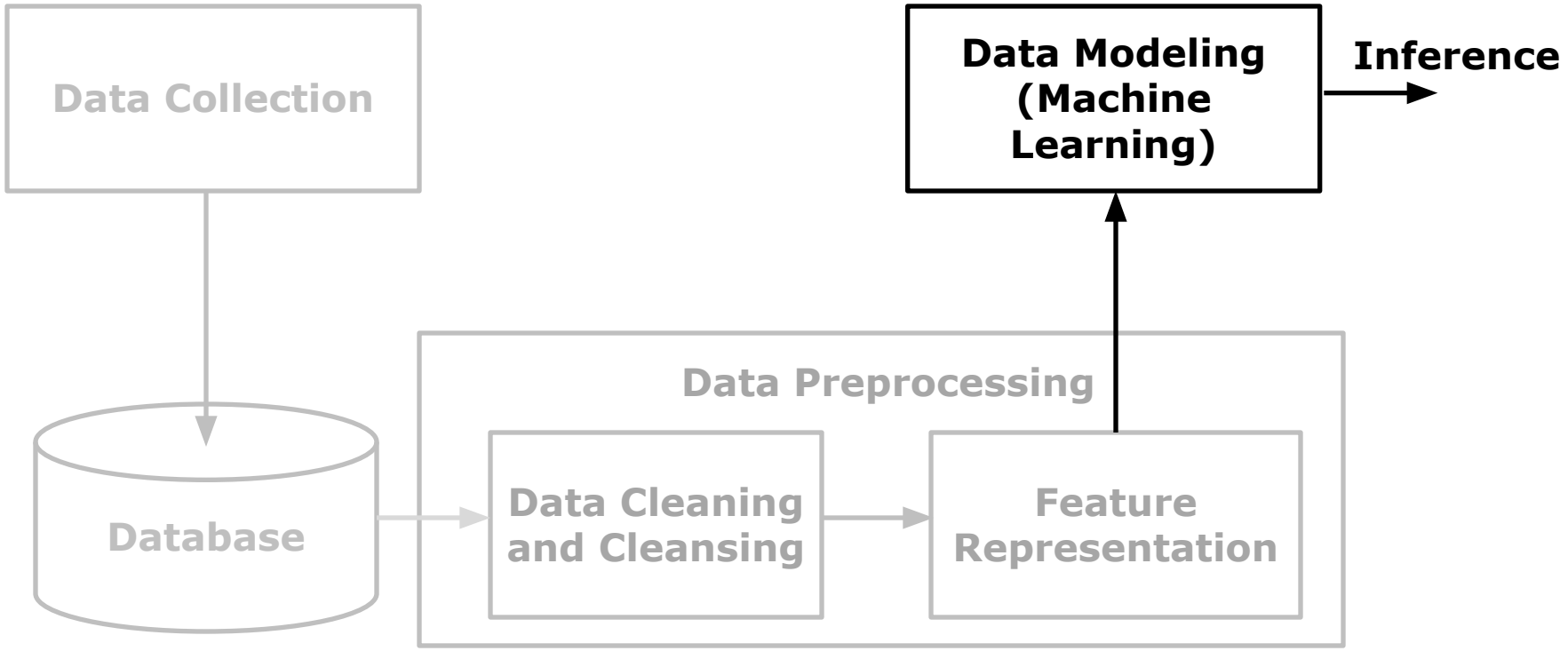


Data Preprocessing and Descriptive Data Analytics

- Data preprocessing involve:
  - Data cleaning, Data integration, Data transformation, Data reduction
- Descriptive data analytics serves as a foundation for data preprocessing
- It helps us to study the general characteristics of data and identify the presence of noise or outliers
- Data characteristics:
  - Central tendency of data
    - Centre of the data
    - Measuring mean, median and mode
  - Dispersion of data
    - The degree to which numerical data tend to spread
    - Measuring range, quartiles, interquartile range (IQR), the five-number summary and standard deviation
- Descriptive analytics are the backbone of reporting

Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data
- Machine learning uses data to extract knowledge – predictive analytics



Predictive Data Analytics

- It is used to identify the trends, correlations and causation by learning the patterns from data
- Study and construction of algorithms that can learn from data and make predictions on data
- It involve tasks like
  - Classification: Categorical label prediction
    - E.g.: predicting the presence or absence of disease or
    - predicting the category of the disease according to symptoms
  - Regression: Numeric prediction
    - E.g.: predicting the amount of landslide or
    - predicting the amount of rainfall
  - Clustering: Grouping of similar patterns
    - E.g.: grouping the similar items to be sold or
    - grouping the people from the same region
- Learning from data

Machine Learning:
Learning from Data

- 1, 2, 3, 4, 5, ?, …, 24, 25, 26, 27, ?
- 1, 3, 5, 7, 9, ?, …, 25, 27, 29, 31, ?
- 2, 3, 5, 7, 11, ?, …, 29, 31, 37, 41, ?
- 1, 4, 9, 16, 25, ?, …, 121, 144, 169, ?
- 1, 2, 4, 8, 16, 32, ?,…, 1024, 2048, 4096, ?
- 1, 1, 2, 3, 5, 8, ?, …, 55, 89, 144, 233, ?
- 1, 1, 2, 4, 7, 13, ?, 44, 81, 149, 274, 504, ?
- 3, 5, 12, 24, 41, ?, …., 201, 248, 300, 357, ?
- 1, 6, 19, 42, 59, ?, …, 95, 117, 156, 191, ?

- 1, 2, 3, 4, 5, 6, …, 24, 25, 26, 27, 28
- 1, 3, 5, 7, 9, 11, …, 25, 27, 29, 31, 33
- 2, 3, 5, 7, 11, 13, …, 29, 31, 37, 41, 43
- 1, 4, 9, 16, 25, 36, …, 121, 144, 169, 196
- 1, 2, 4, 8, 16, 32, 64,…, 1024, 2048, 4096, 8192
- 1, 1, 2, 3, 5, 8, 13, …, 55, 89, 144, 233, 377
- 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927
- 3, 5, 12, 24, 41, 63, ….., 201, 248, 300, 357, 419

  (2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57, 62)
- 1, 6, 19, 42, 59, ?, …, 95, 117, 156, 191, ?

- **Pattern: Any regularity or structure in data or source of data**
- **Pattern Analysis: Automatic discovery of patterns in data**

Image Classification

Tiger

Giraffe

Horse

Bear

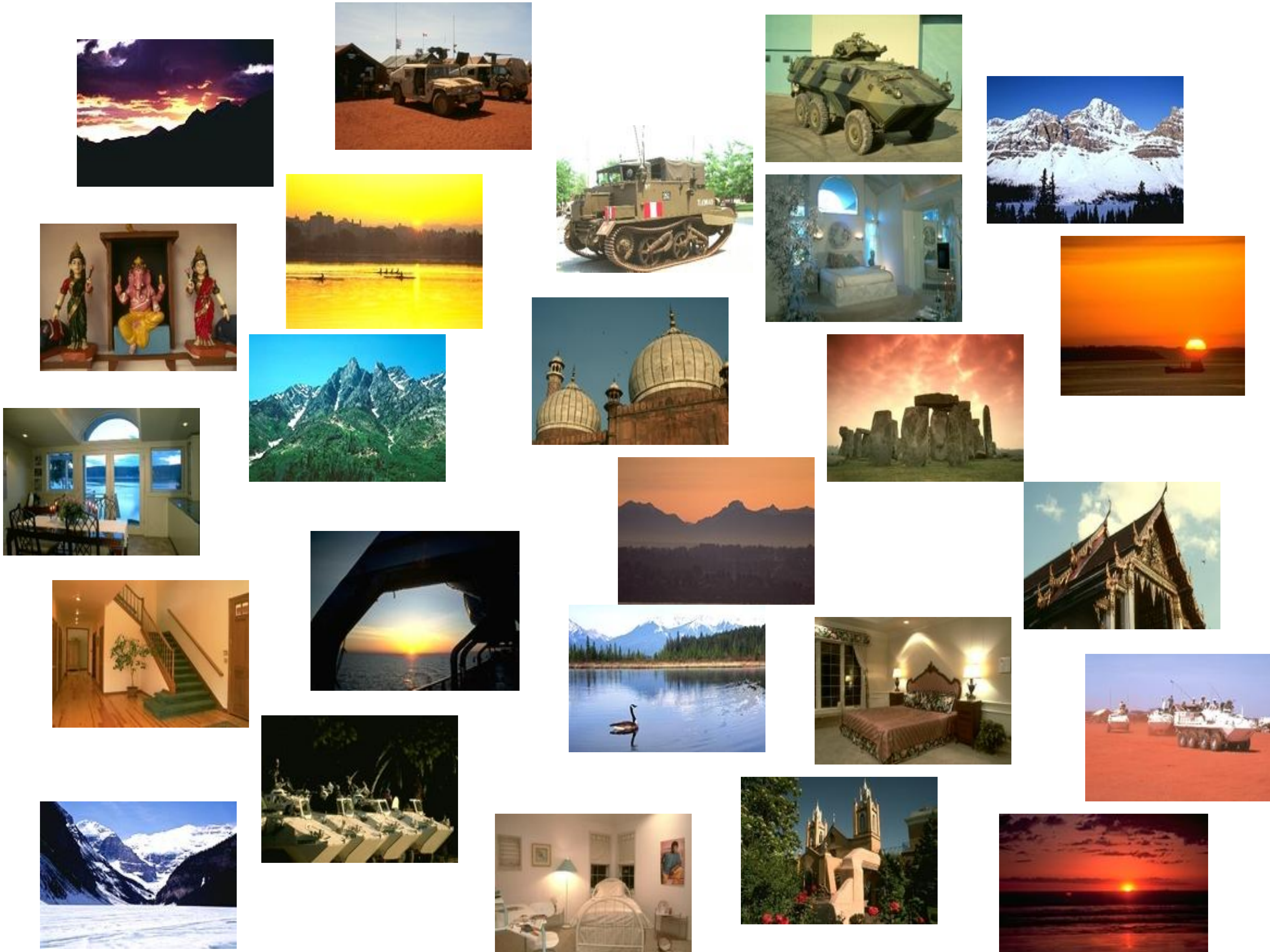
Intraclass variability

Scene Image Classification

Interclass similarity

Tall building	Inside city	Street	Highway	Coast	Open country	Mountain	Forest

Scene Image Clustering



Scene Image Clustering

Residential Interiors

Mountains

Military Vehicles

Sacred Places

Sunsets & Sunrises

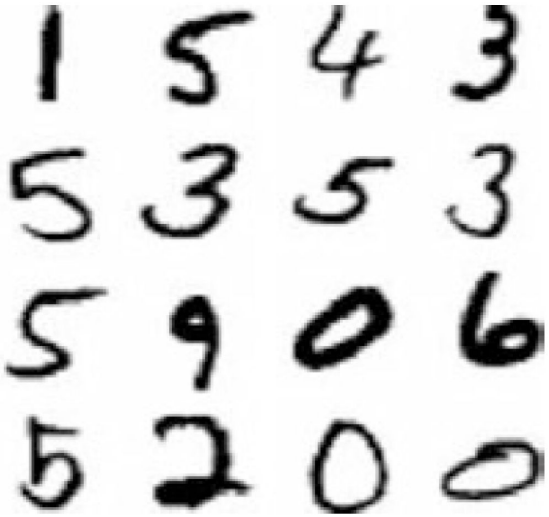
Machine Learning for Pattern Recognition

- Learning: Acquiring new knowledge or modifying the existing knowledge
- Knowledge: Familiarity with information present in data
- Learning by machines for pattern analysis: Acquisition of knowledge from data to discover patterns in data
- Data-driven techniques for learning by machines: Learning from examples (Training of models)
- Generalization ability of learning machines: Performance of trained models on new (test) data
- Target of learning techniques: Good generalization ability
- Learning techniques: Estimation of parameters of models
- Learning machines and Learning techniques for pattern analysis:
  - Statistical Models (Maximum likelihood)
  - Artificial Neural Networks (Error correction learning)
  - Kernel Methods (Learning optimal linear relationships)

Machine Learning Definition

- Arthur Samuel (1959)
  - Field of study that gives computers the ability to learn without being explicitly programmed

- Tom Mitchel (1998)
  - A computer program is said to learn from experience with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience (example) E



- T: Recognizing and classifying handwritten digits presented as image
- P: Percentage of digits correctly classified
- E: Database of handwritten digits images

Machine Learning Definition

- Arthur Samuel (1959)
  - Field of study that gives computers the ability to learn without being explicitly programmed

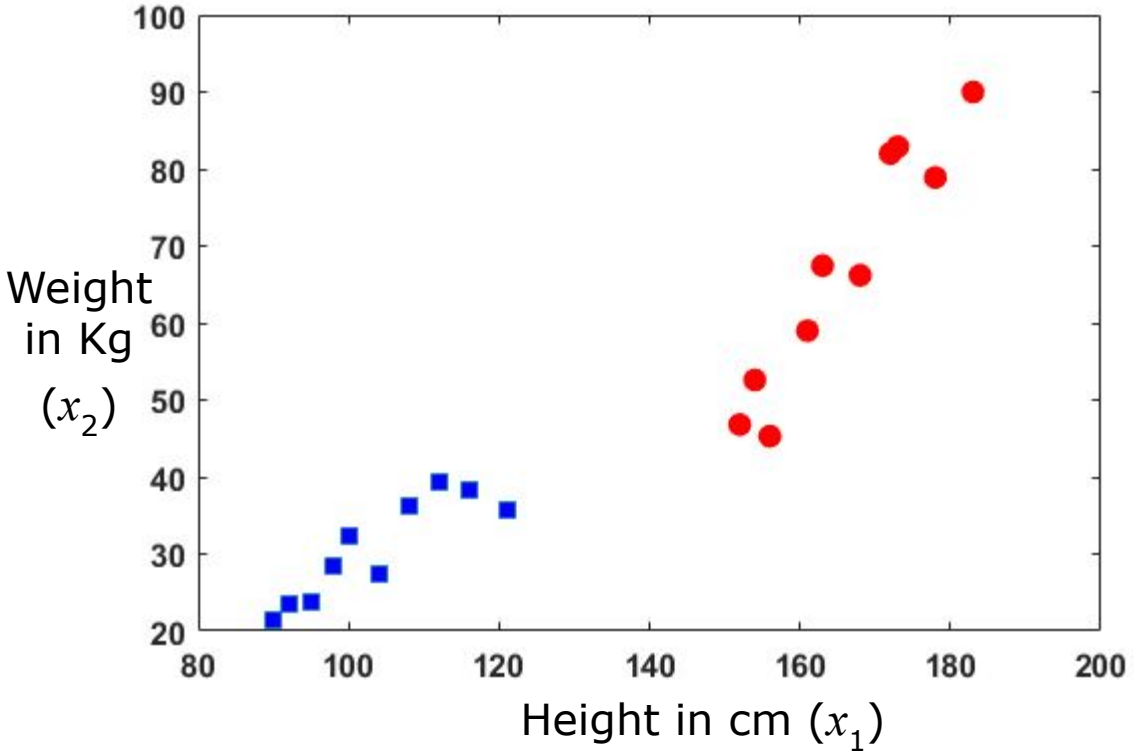
- Tom Mitchel(1998)
  - A computer program is said to learn from experience with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E

- Mapping from input to output

Illustration - Data1: Representing a Person

Height	Weight
90	21.5
95	23.67
100	32.45
116	38.21
98	28.43
108	36.32
104	27.38
112	39.28
121	35.8
92	23.56
152	46.8
178	78.9
163	67.45
173	82.9
154	52.6
168	66.2
183	90
172	82
156	45.3
161	59

- A person is represented using two attributes:
  - Height
  - Weight



$$\mathbf{x} = [x_1 \;\; x_2]^\top$$

Illustration – Data2: Iris (Flower) Data [1]

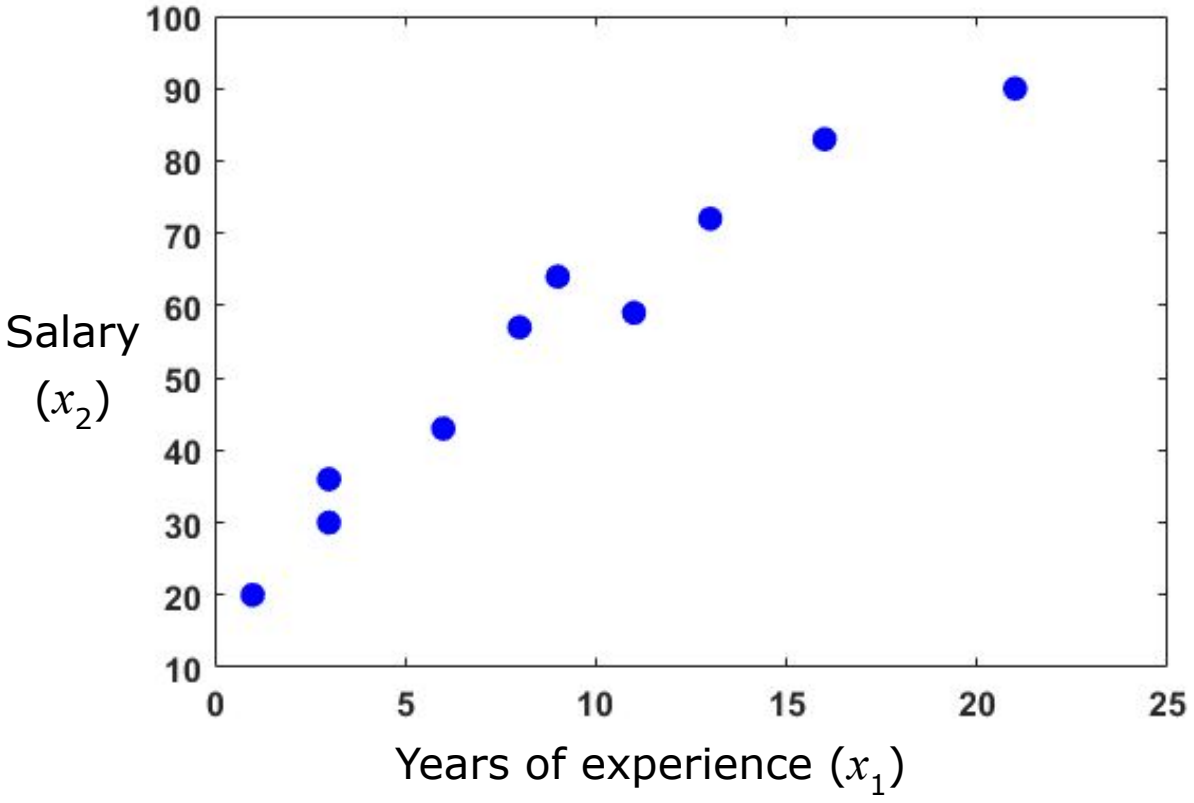
Sepal_Length	Sepal_Width	Petal_Length	Petal_Width
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
4.7	3.2	1.3	0.2
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
6.3	3.3	6	2.5
5.8	2.7	5.1	1.9
7.1	3	5.9	2.1
5.7	2.8	4.1	1.3
7.3	2.9	6.3	1.8
7.3	2.9	6.3	1.8
5.3	3.7	1.5	0.2
4.9	2.4	3.3	1
5	3.5	1.6	0.6
6.3	3.3	4.7	1.6
5.8	2.7	3.9	1.2
5.8	2.8	5.1	2.4
4.4	3	1.3	0.2
6.2	3.4	5.4	2.3

$$\mathbf{x} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4]^\top$$

[1] R. A. Fisher, "*The use of multiple measurements in taxonomic problems*" Annual Eugenics, 7, Part II, pp. 179-188, 1936.

Illustration – Data3: Years of Experience and Salary

Years of experience (x_1)	Salary (in Rs 1000) (x_2)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83



$$\mathbf{x} = [x_1 \;\; x_2]^\top$$

Illustration – Data4: Environmental Data

Temperature	Humidity	Pressure	Rain
25.47	82.19	1036.35	6.75
26.19	83.15	1037.60	1761.75
25.17	85.34	1037.89	652.50
24.30	87.69	1036.86	963.00
24.07	87.65	1027.83	254.25
21.21	95.95	1006.92	339.75
23.49	96.17	1006.57	38.25
21.79	98.59	1009.42	29.25
25.09	88.33	991.65	4.50
25.39	90.43	1009.66	112.50
23.89	94.54	1009.27	735.75
22.51	99.00	1009.80	607.50
22.90	98.00	1009.90	717.75
21.72	99.00	996.29	513.00
23.18	98.97	800.00	195.75
21.24	99.00	1009.21	474.75
21.63	99.00	1008.89	409.50
20.91	99.00	1008.89	1161.00
23.67	97.80	1009.38	0.00
24.53	92.90	1008.66	0.00

$$\mathbf{x} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4]^\top$$

Supervised and Unsupervised Learning

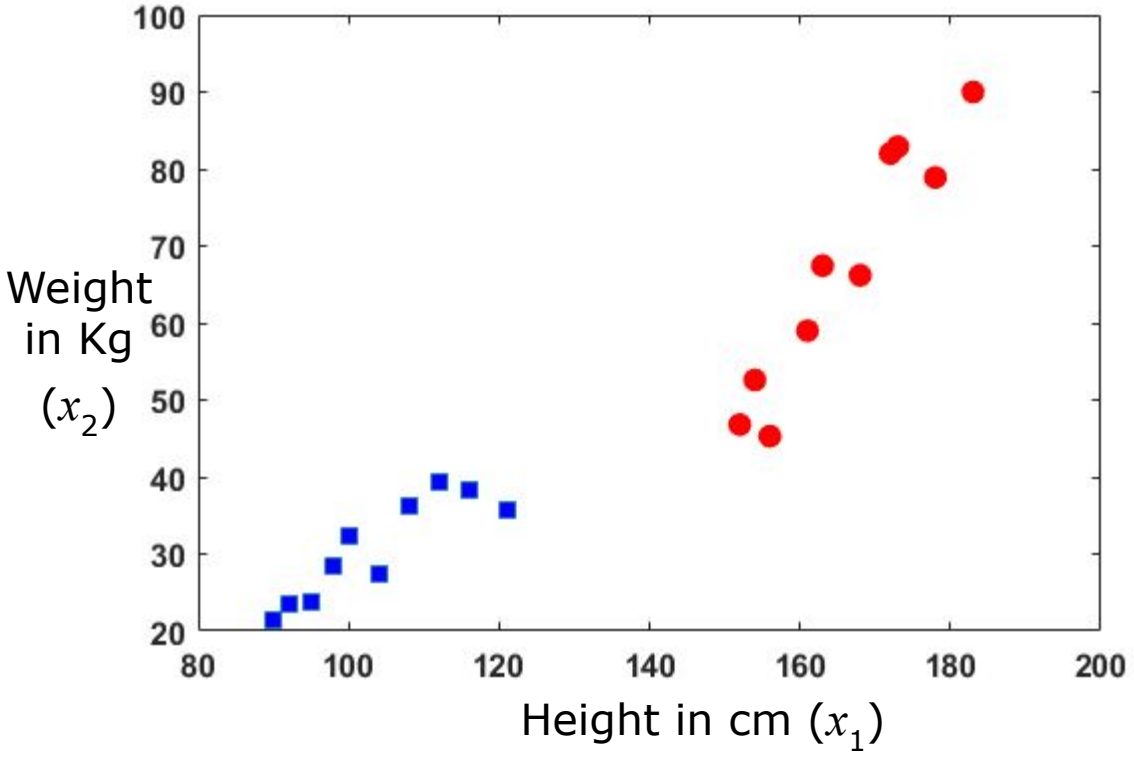
Supervised Learning

- Learning under the supervision
  - Student learning from teacher
  - Child learning to recognize objects/animals
- In the context of machine learning, data used for learning (Train data) is labeled
- Labeled data: Data for which the target value is already known

Labeled Data – Illustration: Data1 - Representing a Person

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- A person is represented using two attributes:
  - Height
  - Weight
- Class (y):
  - Child (0)
  - Adult (1)



$\mathbf{x} = [x_1 \;\; x_2]^\top$ Target/Output : $y \in \{0,1\}$

Labeled Data – Illustration: Data2 - Iris (Flower) Data

Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Class
5.1	3.5	1.4	0.2	1
4.9	3	1.4	0.2	1
4.7	3.2	1.3	0.2	1
7	3.2	4.7	1.4	2
6.4	3.2	4.5	1.5	2
6.9	3.1	4.9	1.5	2
6.3	3.3	6	2.5	3
5.8	2.7	5.1	1.9	3
7.1	3	5.9	2.1	3
5.7	2.8	4.1	1.3	2
7.3	2.9	6.3	1.8	3
7.3	2.9	6.3	1.8	3
5.3	3.7	1.5	0.2	1
4.9	2.4	3.3	1	2
5	3.5	1.6	0.6	1
6.3	3.3	4.7	1.6	2
5.8	2.7	3.9	1.2	2
5.8	2.8	5.1	2.4	3
4.4	3	1.3	0.2	1
6.2	3.4	5.4	2.3	3

- Class (y):
  - Iris Setosa (1)
  - Iris Versicolour (2)
  - Iris Virginica (3)

$$\mathbf{x} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4]^\top$$

Target/Output:

$$y \in \{1, 2, 3\}$$

Labeled Data – Illustration: Data3 - Years of Experience and Salary

Years of experience (x_1)	Salary (in Rs 1000) (x_2)	Raise (y)
3	30	1
8	57	0
9	64	1
13	72	1
3	36	1
6	43	0
11	59	1
21	90	1
1	20	0
16	83	0

- Class – Raise in Salary (y):
  - Yes(1)
  - No (0)

$\mathbf{x} = [x_1 \;\; x_2]^T$ Target/Output : $y \in \{0,1\}$

Labeled Data – Illustration: Data3 - Years of Experience and Salary

Years of experience (x)	Salary (in Rs 1000) (y)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

- Input variable: Years of experience
- Output variable: Salary

Illustration – Data4: Environmental Data

Temperature	Humidity	Pressure	Rain
25.47	82.19	1036.35	6.75
26.19	83.15	1037.60	1761.75
25.17	85.34	1037.89	652.50
24.30	87.69	1036.86	963.00
24.07	87.65	1027.83	254.25
21.21	95.95	1006.92	339.75
23.49	96.17	1006.57	38.25
21.79	98.59	1009.42	29.25
25.09	88.33	991.65	4.50
25.39	90.43	1009.66	112.50
23.89	94.54	1009.27	735.75
22.51	99.00	1009.80	607.50
22.90	98.00	1009.90	717.75
21.72	99.00	996.29	513.00
23.18	98.97	800.00	195.75
21.24	99.00	1009.21	474.75
21.63	99.00	1008.89	409.50
20.91	99.00	1008.89	1161.00
23.67	97.80	1009.38	0.00
24.53	92.90	1008.66	0.00

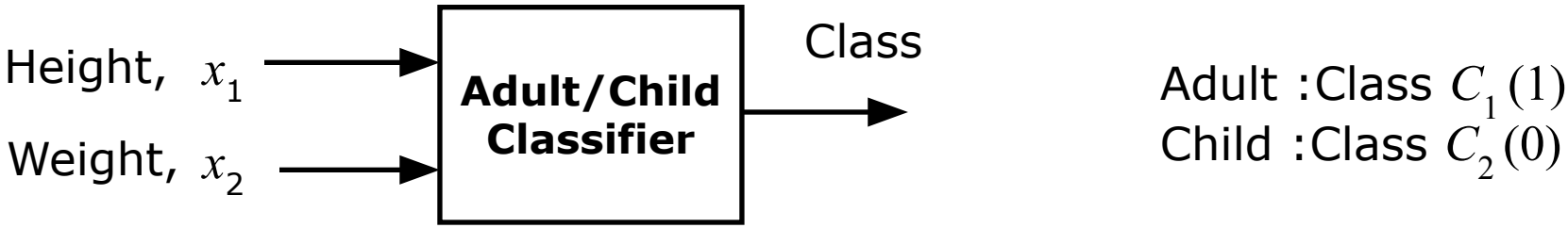
- Predicting `Rain` (target attribute) based on `Temperature`, `Humidity` and `Pressure`
- Input variable: `Temperature`, `Humidity` and `Pressure`
- Output variable: `Rain`

Supervised Learning

- In supervised learning, each example (data sample) is a pair consisting of an input example (typically a vector) and a desired output value (also called the target)
- Task of learning a function that maps an input to an output based on example input-output pairs

$$y_n = f(\mathbf{x}_n)$$

- A supervised learning algorithm
  - analyzes the training data and
  - produces an inferred function, which can be used for predicting the output of a new examples
- One of the scenario will be the algorithm to determine the class labels for unseen instances



Supervised Learning

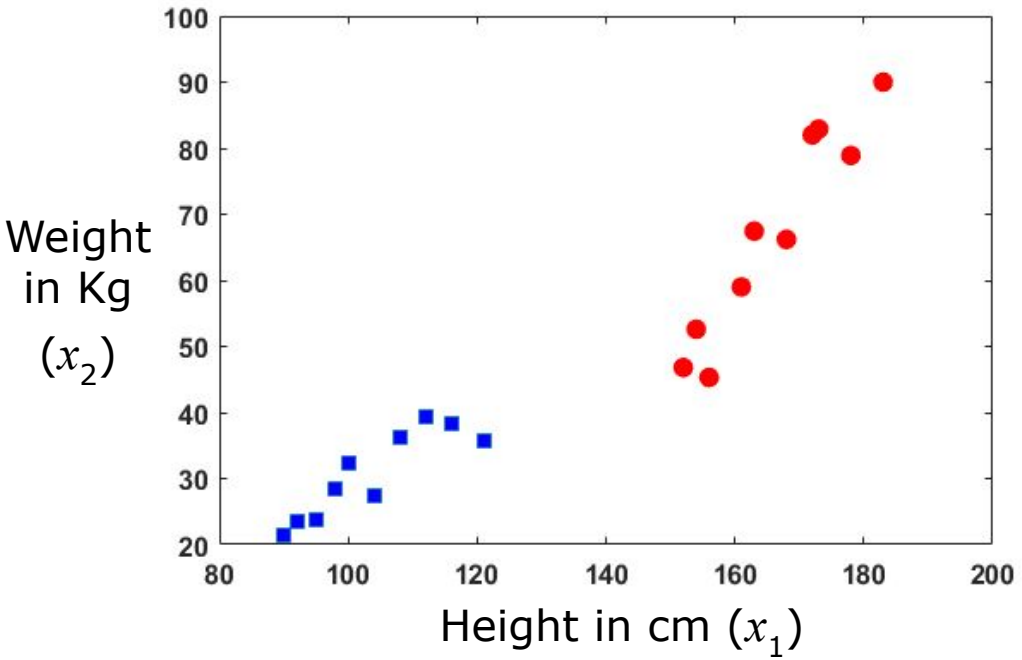
- Supervised learning is grouped into
  - Classification
  - Regression
- Classification:
  - Output variable is categorical
  - Categorical label prediction
  - Example:
    - Predicting a person as adult or child (2-class)
    - Predicting the raise in salary based on the year of experience and salary (2-class)
    - Identify an email as spam or not (2-class)
    - Predicting the presence or absence of disease (2-class)
    - Categorising the disease according to symptoms (Multi-class)
    - Categorizing the Iris flowers (Multi-class)

Supervised Learning

- Supervised learning is grouped into
  - Classification
  - Regression
- Regression:
  - Output variable is real or continuous value
  - Numeric prediction
  - Example:
    - predicting the salary based on the experience
    - predicting the amount of rainfall based on atmospheric temperature, humidity, pressure, amount of sunlight etc.

Unsupervised Learning

- Learning without a supervision
- In the context of machine learning, data used for learning (Train data) is unlabeled
- Given these unlabeled data machine tries to identify the pattern and give the response
- Example:
  - A person is represented using two attributes: Height and Weight
  - No label is given
  - Machine try to learn the patterns from the given set and groups them based on the similarity



Unsupervised Learning

- Unsupervised learning is grouped into
  - Clustering
  - Association
- Clustering:
  - Partitioning the data into cohesive groups such that the data samples in a group are similar
  - Example:
    - Grouping the persons based on their height and weight
    - Given the customer and their purchase data:
      - Grouping the customers based on the similar products purchased
- Association:
  - It is a rule-based machine learning to discover the interesting variables in a data set
  - Example:
    - Given the customer and their purchase data:
      - Finding the products purchased together

Summary

- Machine learning: Learning from data
- Supervised machine learning
  - Data used for learning (Train data) is labeled
  - Each example (data sample) is a pair consisting of an input example (typically a vector) and a desired output value (also called the target)
  - Task of learning a function that maps an input to an output based on example input-output pairs
  - Classification and Regression
- Unsupervised machine learning
  - Data used for learning (Train data) is unlabeled
  - Given these unlabeled data machine tries to identify the pattern based on similarity
  - Clustering and Association

Text Books

1. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann Publishers, 2011.

2. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 2009.

3. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

Supervised Machine Learning: Pattern Classification

Classification

- Problem of identifying to which of a set of categories a new observation belongs
- Predicts categorical labels
- Example:
  - Predicting a person as adult or child (2-class)
  - Predicting the raise in salary based on the year of experience and salary (2-class)
  - Identify an email as spam or not (2-class)
  - Predicting the presence or absence of disease (2-class)
    - **Pima Indians Diabetes Database:** predict whether a patient has diabetes or not based on diagnostic measurements
  - Categorising the disease according to symptoms (Multi-class)
  - Categorizing the Iris flowers (Multi-class)

Classification

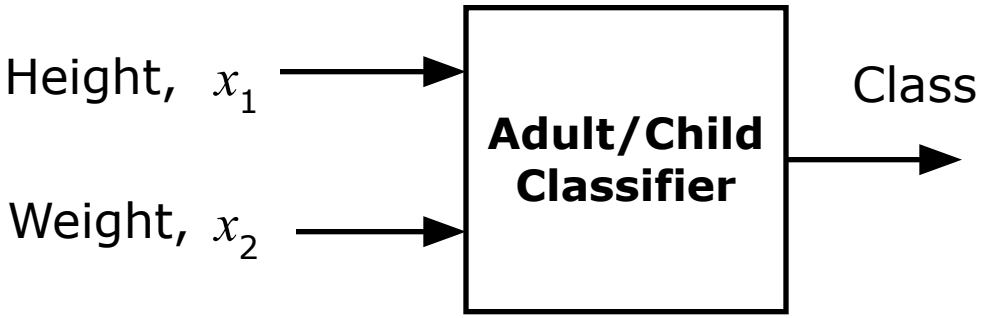
- Classification is a two step process
  - Step1: Building a classifier (data modeling)
    - Learning from data (training phase)
    - Supervised learning: In supervised learning, each example is a *pair* consisting of an input example and a desired output value (class label)
    - Training phase or learning phase is viewed as the learning of a mapping or function that can predict the associated class label of a given training example

$$y_n = f(\mathbf{x}_n)$$

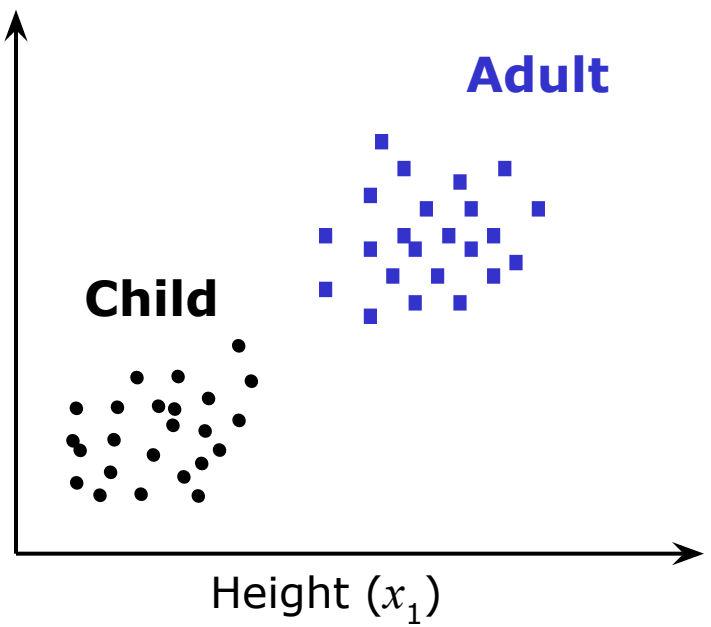
      - $\mathbf{x}_n$ is the n^{th} training example and y_n is the associated class label
  - Step2: Using classification model for prediction
    - Testing phase - Predicting class label for the unseen data
- Accuracy of a classifier: Percentage of test examples that are correctly classified by the classifier
- Target of learning techniques: Good generalization ability

2-class Classification

- Example: Classifying a person as child or adult



Adult :Class C_1
Child :Class C_2

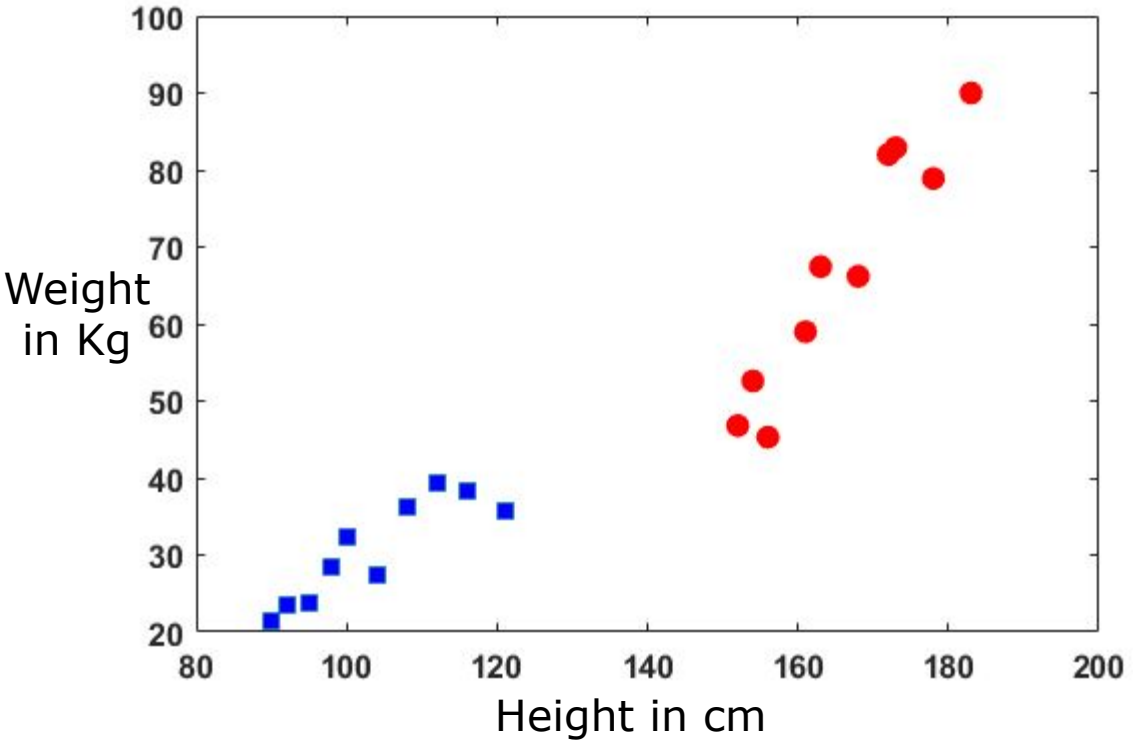


$$\mathbf{x} = [x_1 \;\; x_2]^T$$

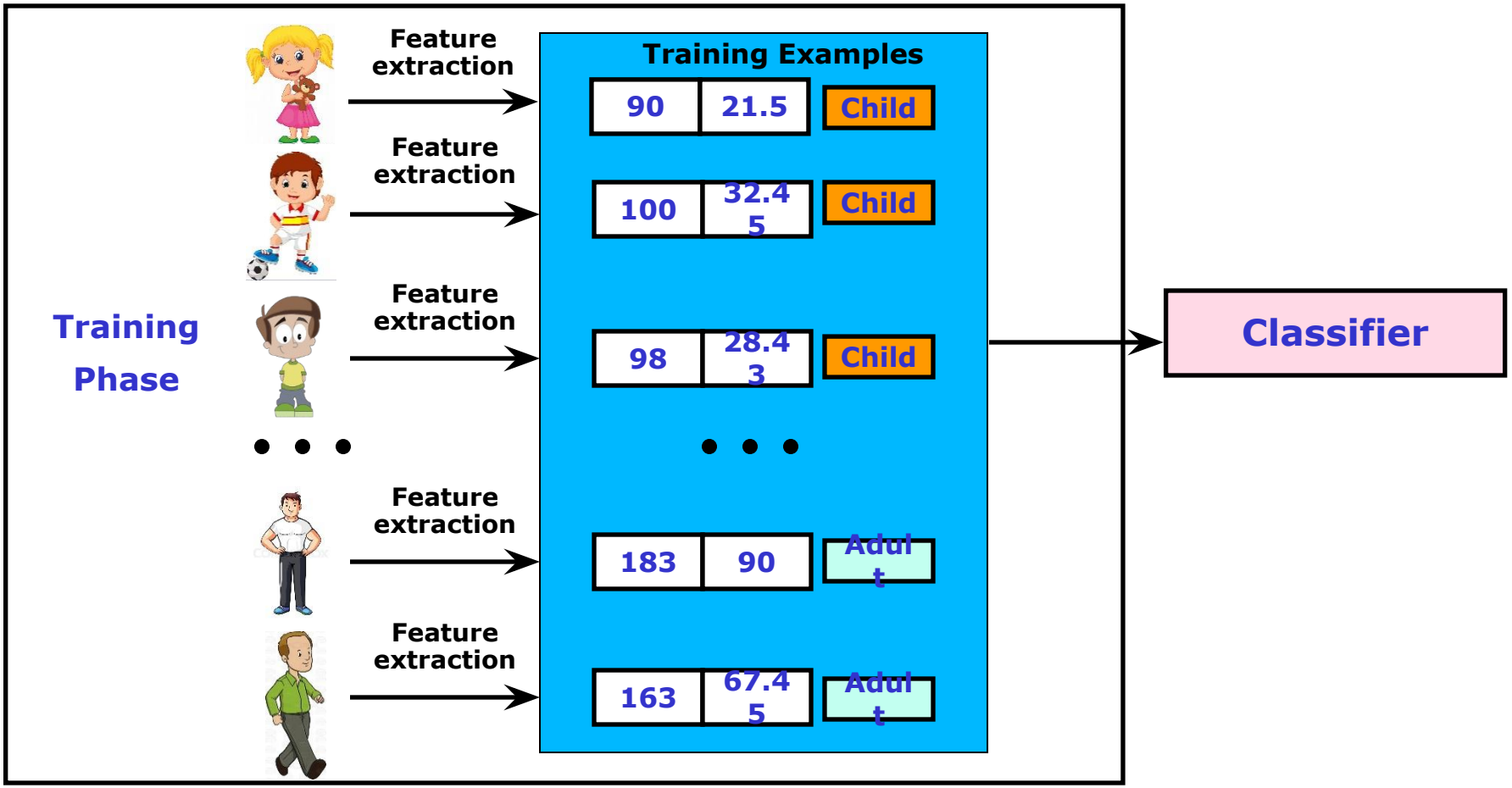
Illustration of Training Set: Adult-Child

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

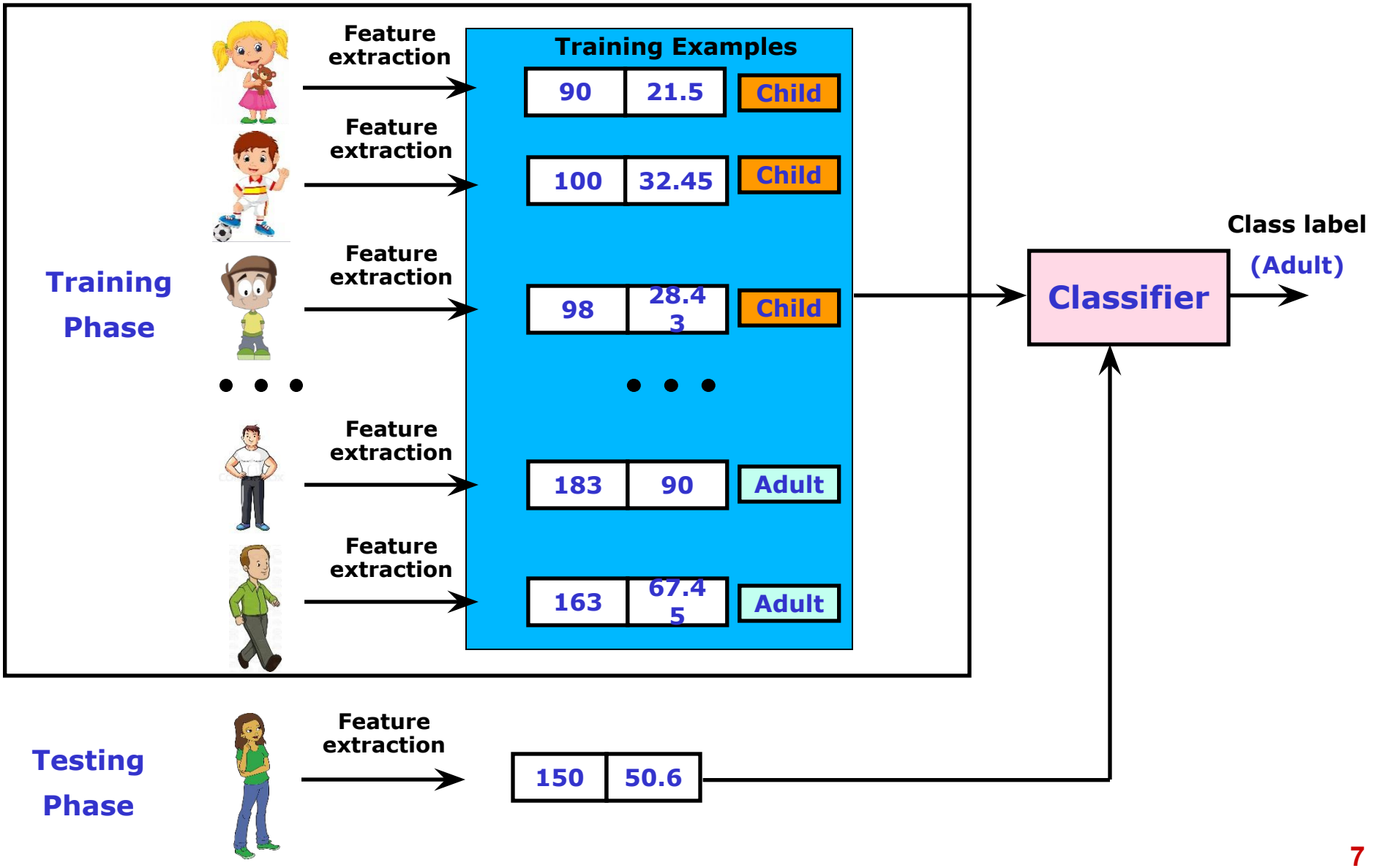
- Number of training examples (N) = 20
- Dimension of a training example = 2
- Class label attribute is 3rd dimension
- Class:
  - Child (0)
  - Adult (1)



Step1: Building a Classification Model (Training Phase)



Step2: Classification (Testing Phase)



Data Preparation for the Classification

- Divide the data into training set and test set
- **Approach 1:** When the number samples from each class are almost equal (Balanced data)
  - Most common split is 70-30 split:
    - Training data contain 70% of samples from each class
    - Test data contain remaining 30% of samples from each class
  - One can use other splits like 50-50 or 60-40 or 80-20 or 90-10

Data Preparation for the Classification: Approach 1

- Suppose that we are doing 70-30 split
- Suppose the data set has 3000 samples
- Each sample is belonging to one of the 3 classes
- Suppose each class has 1000 samples
  - Step1: From class1, 70% i.e. 700 samples considered as training samples and remaining 30% i.e. 300 samples are considered as test samples
  - Step2: From class2, 70% i.e. 700 samples considered as training samples and remaining 30% i.e. 300 samples are considered as test samples
  - Step3: From class3, 70% i.e. 700 samples considered as training samples and remaining 30% i.e. 300 samples are considered as test samples
  - Step4: Combine training examples from each class
    - Training set now contain 700+700+700=2100 samples
  - Step5: Combine test examples from each class
    - Test set now contain 300+300+300=900 samples

Data Preparation for the Classification

- Divide the data into training set and test set
- **Approach 1:** When the number samples from each class are almost equal (Balanced data)
  - Example:
    - Training data contain 70% of samples from each class
    - Test data contain remaining 30% of samples from each class
- **Approach 2:** When the number samples from each class are not equal (Imbalanced data)
  - One class may have large number of samples and another has small number of samples
  - 70%-30% division may cause learned model to be bias to class with larger number of training samples
  - Solution:
    - Consider 70% or 80% of the samples from the class with least number of samples as training data from that class
    - Consider the same number of samples from other class as training examples
    - Each class will have same number of training examples

Data Preparation for the Classification: Approach 2

- Suppose the data set has 3000 samples
- Each sample is belonging to one of the 3 classes
- Suppose class1 has 700 samples, class2 has 300 samples and class3 has 2000 samples
  - Step1: From class2, 70% i.e. 210 samples considered as training samples and remaining 30% i.e. 90 samples are considered as test samples
  - Step2: From class1, 210 samples considered as training samples and remaining 490 samples are considered as test samples
  - Step3: From class3, 210 samples considered as training samples and remaining 1790 samples are considered as test samples
  - Step4: Combine training examples from each class
    - Training set now contain 210+210+210=630 samples
  - Step5: Combine test examples from each class
    - Test set now contain 490+90+1790=2370 samples

Nearest-Neighbour Method

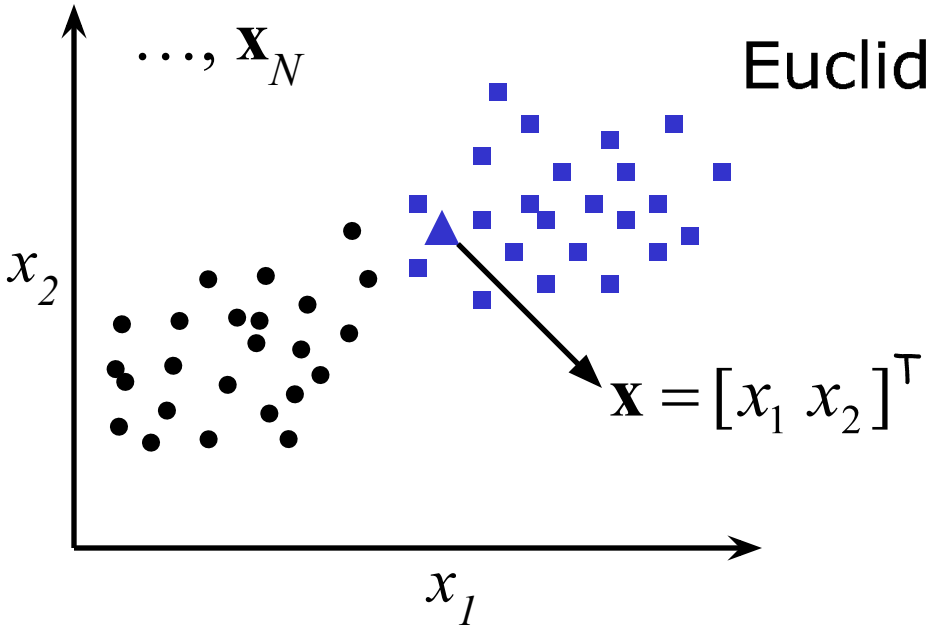
- Training data with N samples: $\mathrm{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$,

  $\mathbf{x}_n \in \mathrm{R}^d$ and $y_n \in \{1, 2, \ldots, M\}$

  – d: dimension of input example

  – M: Number of classes

- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$



$$\text{Euclidean distance} = \|\mathbf{x}_n - \mathbf{x}\|$$

$$= \sqrt{(\mathbf{x}_n - \mathbf{x})^\mathsf{T}(\mathbf{x}_n - \mathbf{x})}$$

$$= \sqrt{\sum_{i=1}^{d}(x_{ni} - x_i)^2}$$

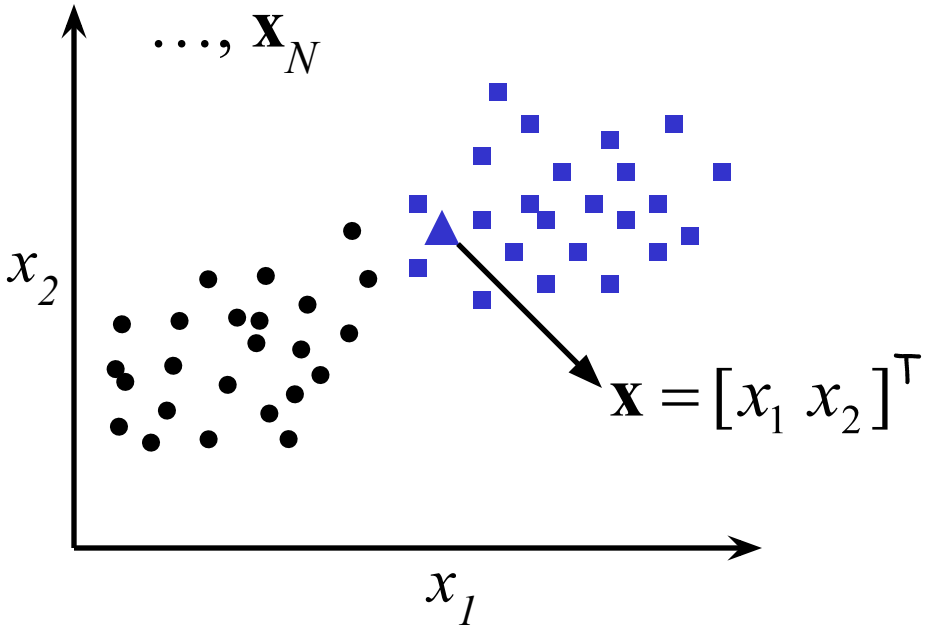
Nearest-Neighbour Method

- Training data: $D = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$,

$$\mathbf{x}_n \in \mathrm{R}^{d} \text{ and } y_n \in \{1, 2, \ldots, M\}$$

  - d: dimension of input example
  - M: Number of classes

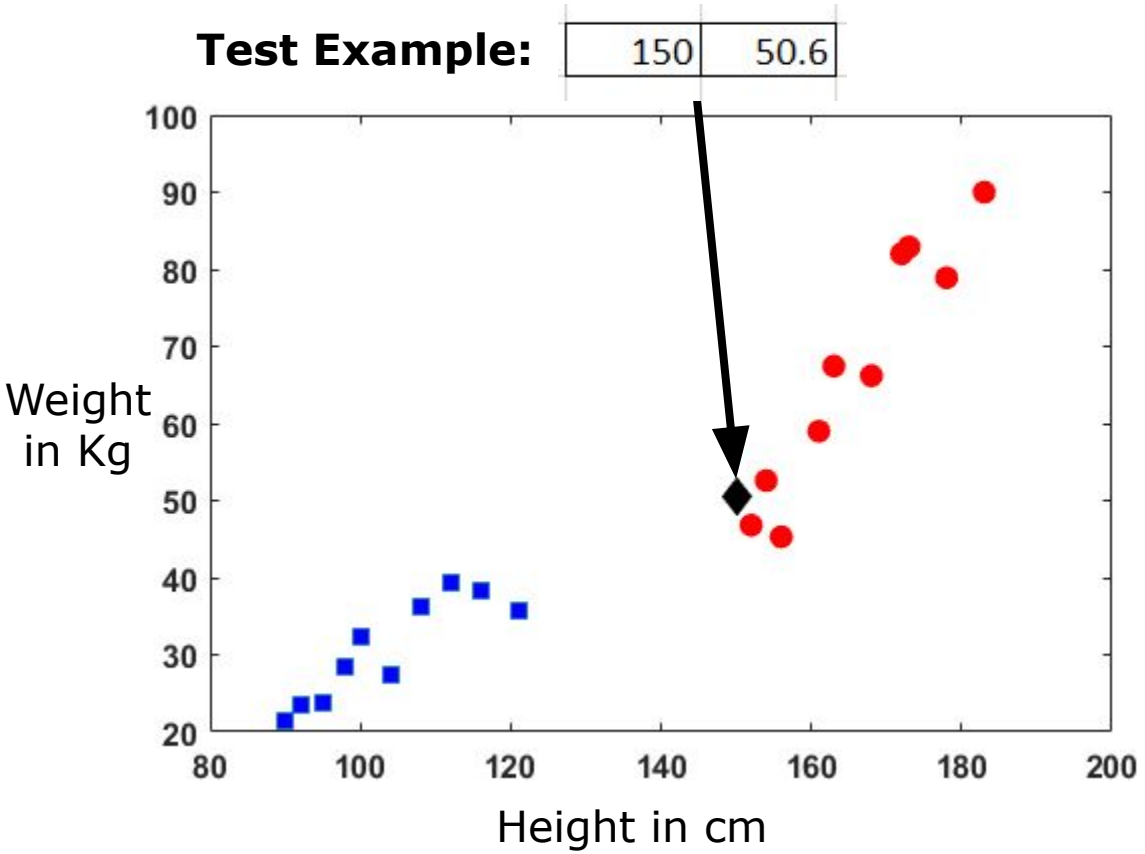
- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$



- Step 2: Sort the examples in the training set in the ascending order of the distance to test example $\mathbf{x}$
- Step 3: Assign the class of the training example with the minimum distance to the test example, $\mathbf{x}$

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

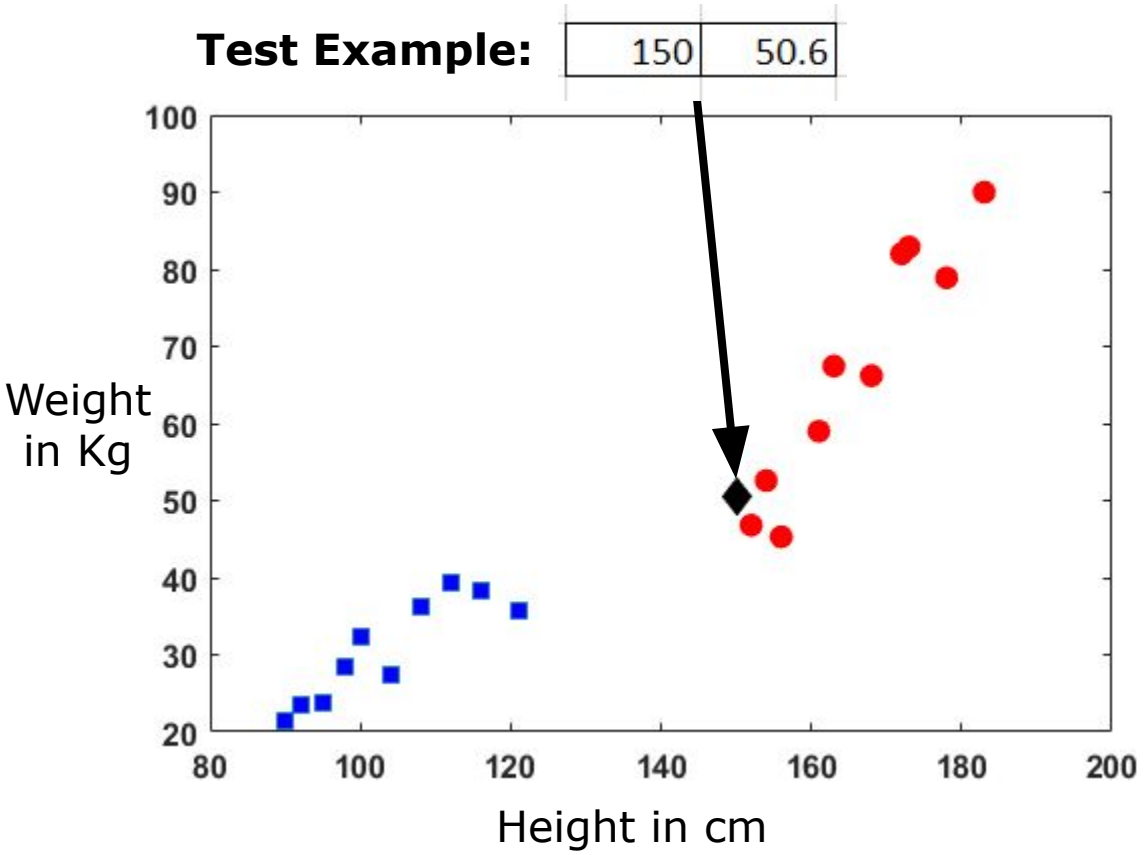
Height	Weight	ED
90	21.5	66.68
95	23.67	61.24
100	32.45	53.19
116	38.21	36.19
98	28.43	56.53
108	36.32	44.36
104	27.38	51.53
112	39.28	39.65
121	35.8	32.56
92	23.56	63.99
152	46.8	4.294
178	78.9	39.81
163	67.45	21.28
173	82.9	39.65
154	52.6	4.472
168	66.2	23.82
183	90	51.39
172	82	38.34
156	45.3	8.006
161	59	13.84



- Step 1: Compute Euclidean distance (ED) will each training examples

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

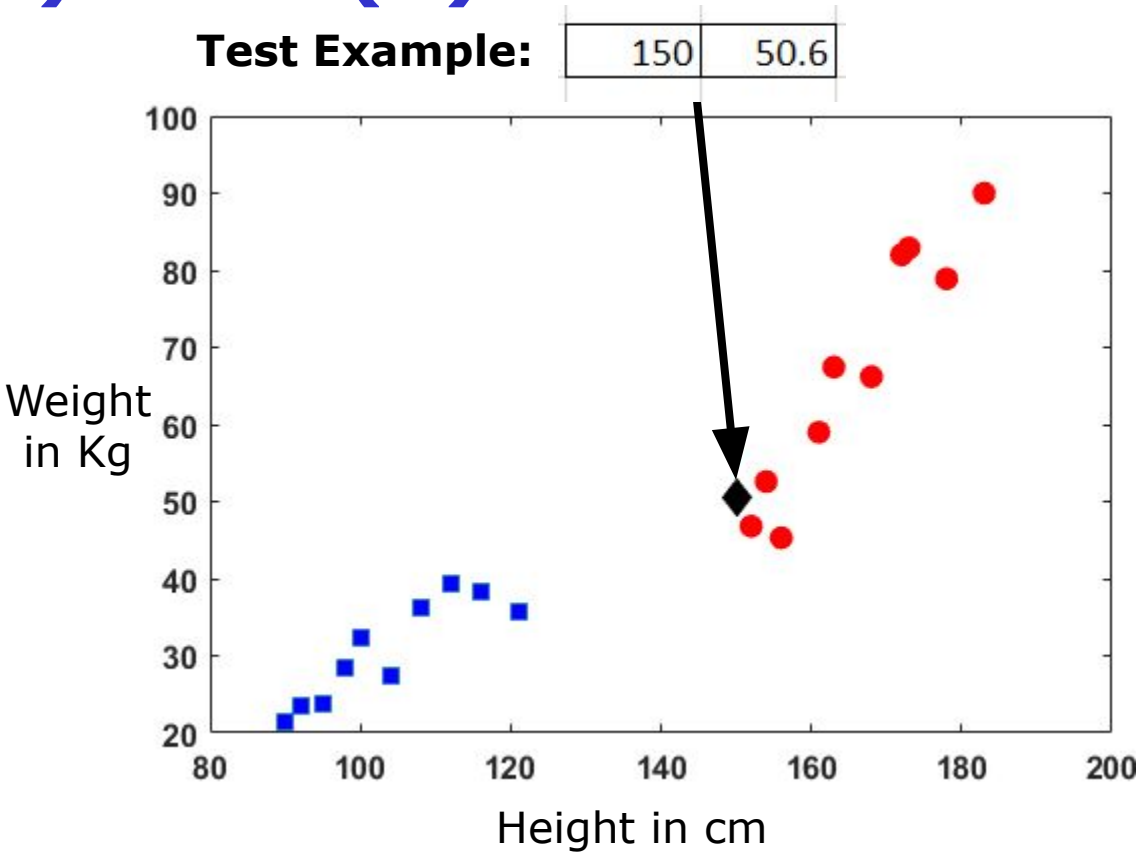
Height	Weight	ED
90	21.5	66.68
95	23.67	61.24
100	32.45	53.19
116	38.21	36.19
98	28.43	56.53
108	36.32	44.36
104	27.38	51.53
112	39.28	39.65
121	35.8	32.56
92	23.56	63.99
152	46.8	4.294
178	78.9	39.81
163	67.45	21.28
173	82.9	39.65
154	52.6	4.472
168	66.2	23.82
183	90	51.39
172	82	38.34
156	45.3	8.006
161	59	13.84



- Step 2: Sort the examples in the training set in the ascending order of the distance to test example

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Class	Height	Weight	ED
0	90	21.5	66.68
0	95	23.67	61.24
0	100	32.45	53.19
0	116	38.21	36.19
0	98	28.43	56.53
0	108	36.32	44.36
0	104	27.38	51.53
0	112	39.28	39.65
0	121	35.8	32.56
0	92	23.56	63.99
1	152	46.8	4.294
1	178	78.9	39.81
1	163	67.45	21.28
1	173	82.9	39.65
1	154	52.6	4.472
1	168	66.2	23.82
1	183	90	51.39
1	172	82	38.34
1	156	45.3	8.006
1	161	59	13.84



- Step 3: Assign the class of the training example with the minimum distance to the test example
  - Class: Adult (1)

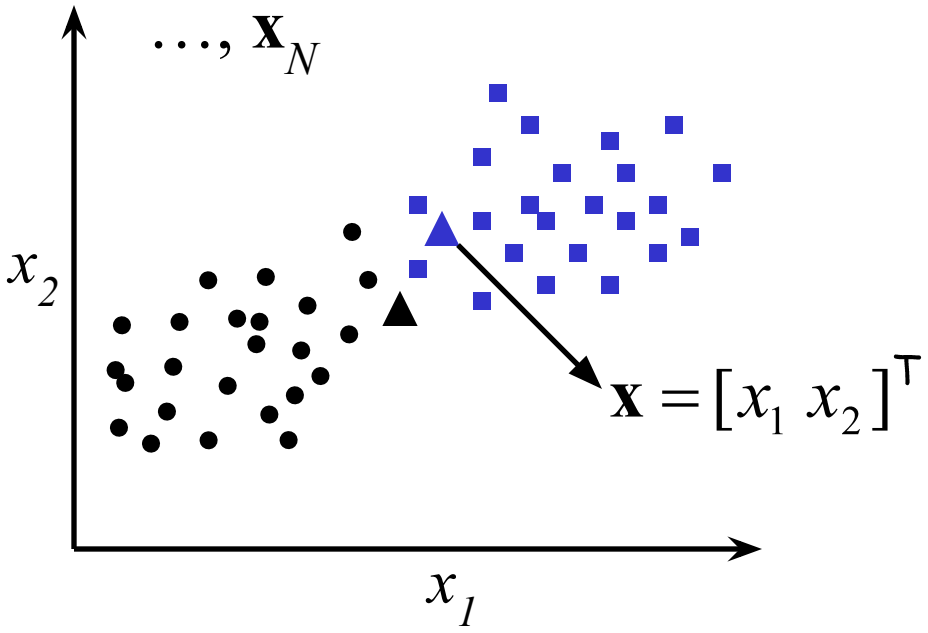
Nearest-Neighbour Method

- Training data: $D = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$,

$$\mathbf{x}_n \in \mathrm{R}^d \text{ and } y_n \in \{1, 2, \ldots, M\}$$

  - d: dimension of input example
  - M: Number of classes

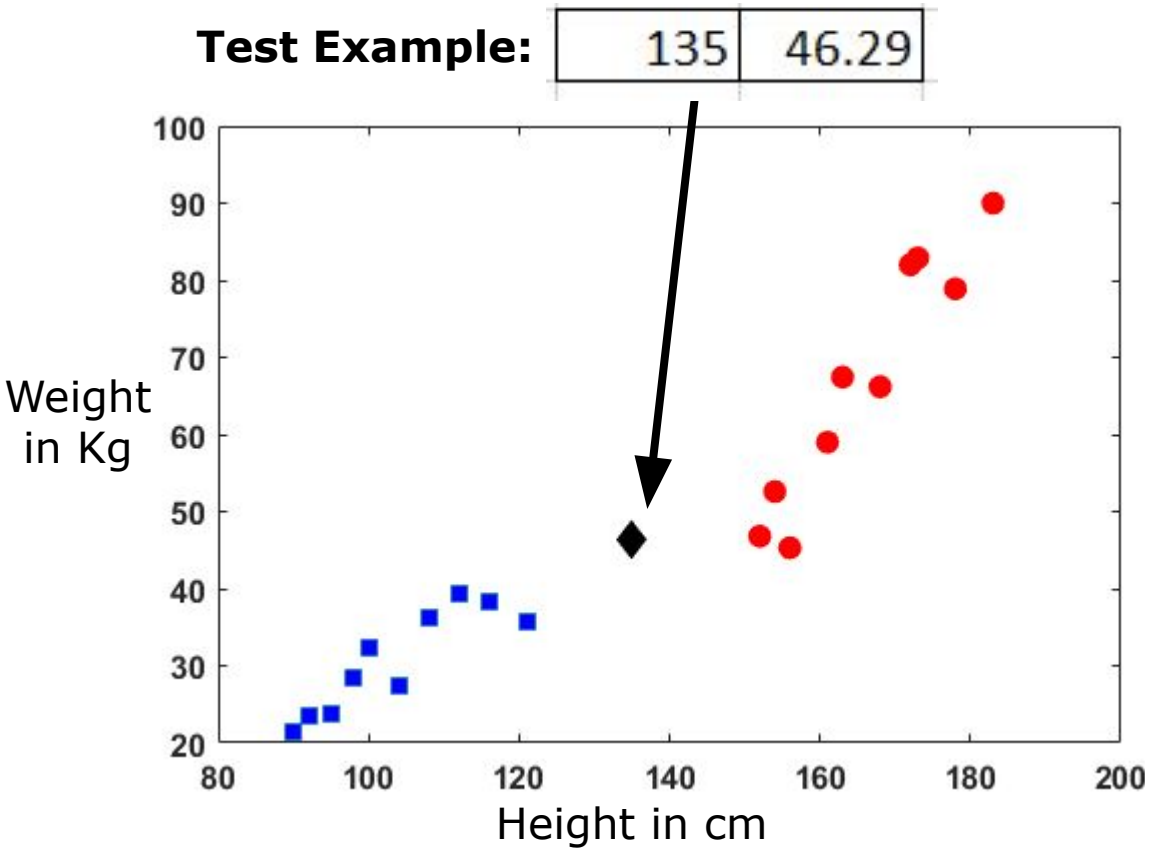
- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$



- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Assign the class of the training example with the minimum distance to the test example, $\mathbf{x}$

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

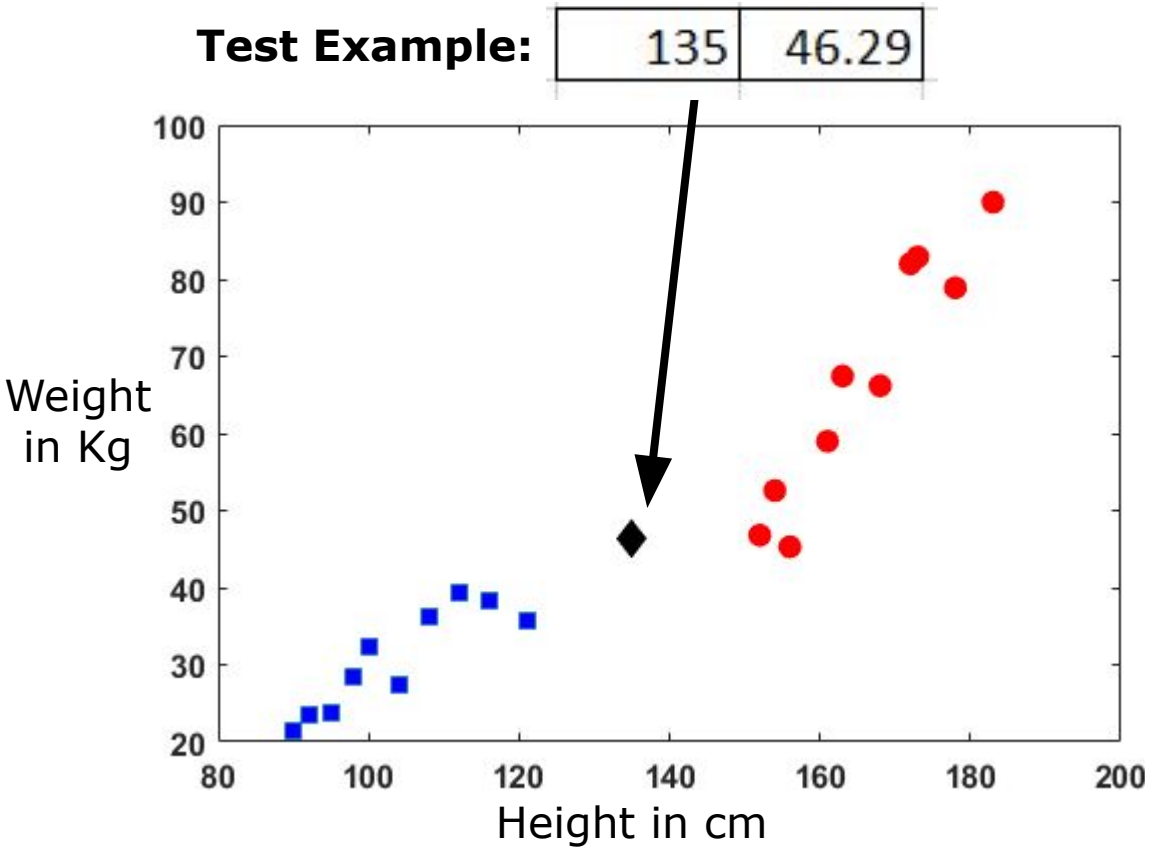
Height	Weight	ED
90	21.5	51.38
95	23.67	45.95
100	32.45	37.64
116	38.21	20.65
98	28.43	41.09
108	36.32	28.78
104	27.38	36.31
112	39.28	24.04
121	35.8	17.49
92	23.56	48.64
152	46.8	17.01
178	78.9	53.97
163	67.45	35.1
173	82.9	52.77
154	52.6	20.02
168	66.2	38.54
183	90	64.92
172	82	51.42
156	45.3	21.02
161	59	28.94



- Step 1: Compute Euclidean distance (ED) will each training examples

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

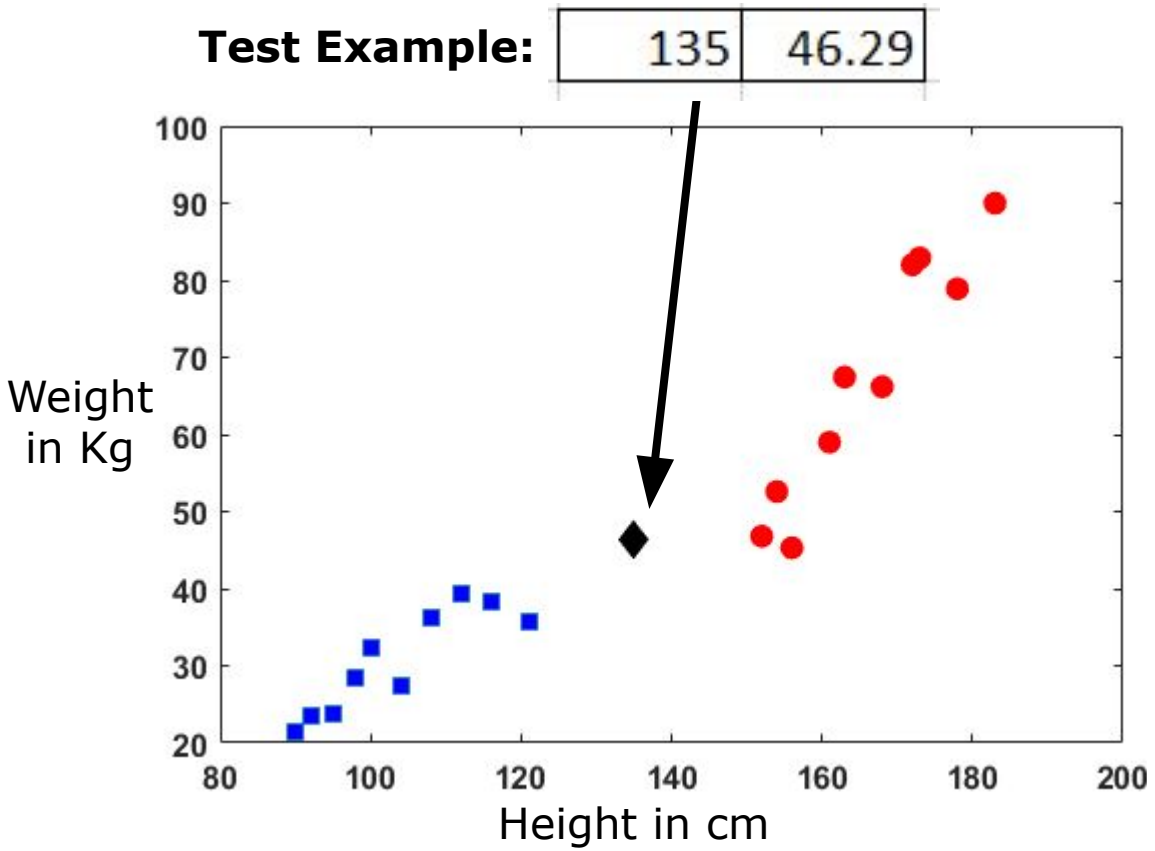
Height	Weight	ED
90	21.5	51.38
95	23.67	45.95
100	32.45	37.64
116	38.21	20.65
98	28.43	41.09
108	36.32	28.78
104	27.38	36.31
112	39.28	24.04
121	35.8	17.49
92	23.56	48.64
152	46.8	17.01
178	78.9	53.97
163	67.45	35.1
173	82.9	52.77
154	52.6	20.02
168	66.2	38.54
183	90	64.92
172	82	51.42
156	45.3	21.02
161	59	28.94



- Step 2: Sort the examples in the training set in the ascending order of the distance to test example

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

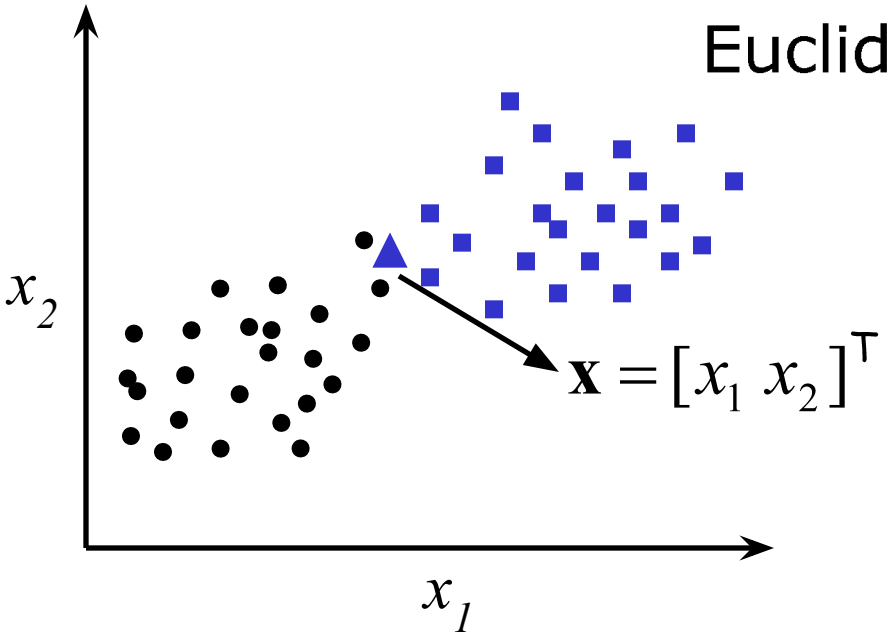
Class	Height	Weight	ED
0	90	21.5	51.38
0	95	23.67	45.95
0	100	32.45	37.64
0	116	38.21	20.65
0	98	28.43	41.09
0	108	36.32	28.78
0	104	27.38	36.31
0	112	39.28	24.04
0	121	35.8	17.49
0	92	23.56	48.64
1	152	46.8	17.01
1	178	78.9	53.97
1	163	67.45	35.1
1	173	82.9	52.77
1	154	52.6	20.02
1	168	66.2	38.54
1	183	90	64.92
1	172	82	51.42
1	156	45.3	21.02
1	161	59	28.94



- Step 3: Assign the class of the training example with the minimum distance to the test example
  - Class: Adult (1) ?

K-Nearest Neighbours (K-NN) Method

- Consider the class labels of the K training examples nearest to the test example
- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$



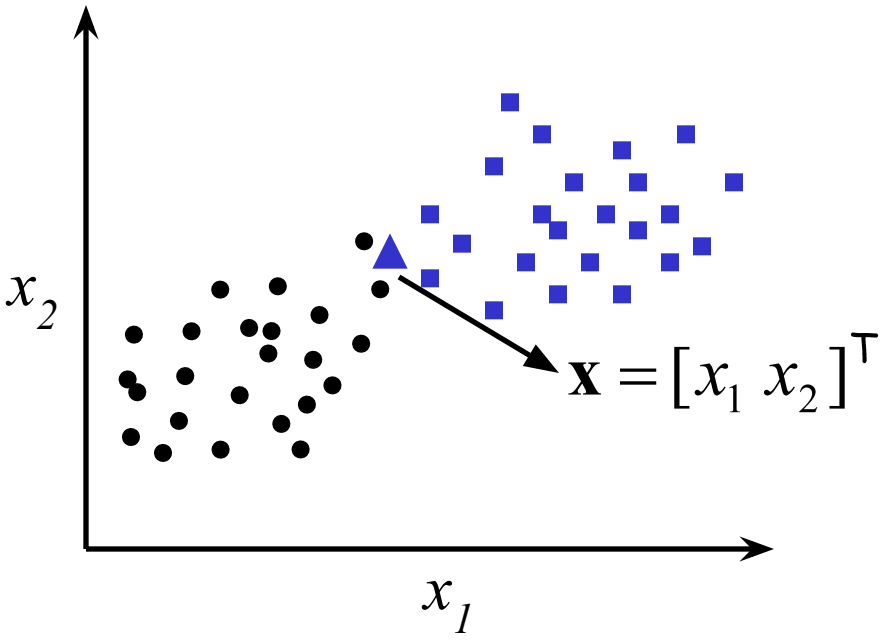
$$\text{Euclidean distance} = \|\mathbf{x}_n - \mathbf{x}\|$$

$$= \sqrt{(\mathbf{x}_n - \mathbf{x})^T (\mathbf{x}_n - \mathbf{x})}$$

$$= \sqrt{\sum_{i=1}^{d} (x_{ni} - x_i)^2}$$

K-Nearest Neighbours (K-NN) Method

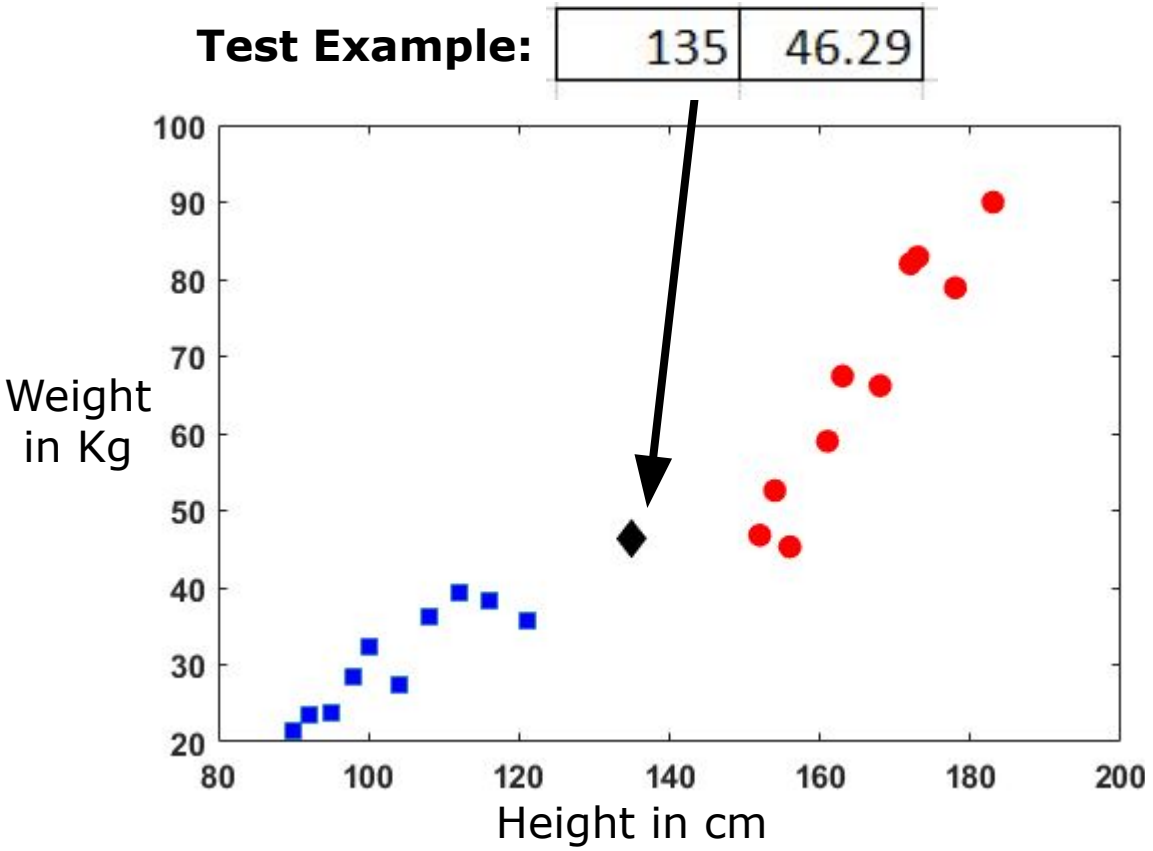
- Consider the class labels of the K training examples nearest to the test example
- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$



- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Choose the first K examples in the sorted list
  - K is the number of neighbours for text example
- Step 4: Test example is assigned the most common class among its K neighbours

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

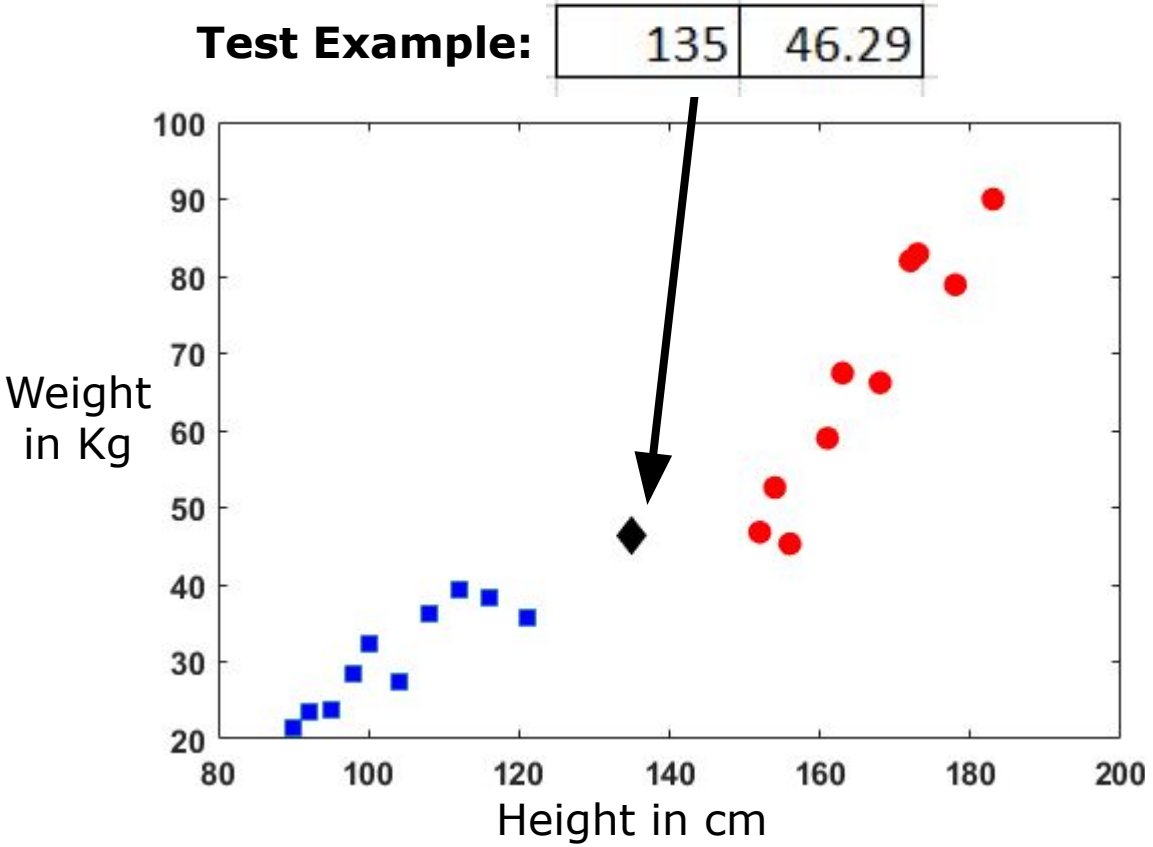
Class	Height	Weight	ED
0	90	21.5	51.38
0	95	23.67	45.95
0	100	32.45	37.64
0	116	38.21	20.65
0	98	28.43	41.09
0	108	36.32	28.78
0	104	27.38	36.31
0	112	39.28	24.04
0	121	35.8	17.49
0	92	23.56	48.64
1	152	46.8	17.01
1	178	78.9	53.97
1	163	67.45	35.1
1	173	82.9	52.77
1	154	52.6	20.02
1	168	66.2	38.54
1	183	90	64.92
1	172	82	51.42
1	156	45.3	21.02
1	161	59	28.94



- Consider K=5
- Step 3: Choose the first K=5 examples in the sorted list

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Class	Height	Weight	ED
0	90	21.5	51.38
0	95	23.67	45.95
0	100	32.45	37.64
0	116	38.21	20.65
0	98	28.43	41.09
0	108	36.32	28.78
0	104	27.38	36.31
0	112	39.28	24.04
0	121	35.8	17.49
0	92	23.56	48.64
1	152	46.8	17.01
1	178	78.9	53.97
1	163	67.45	35.1
1	173	82.9	52.77
1	154	52.6	20.02
1	168	66.2	38.54
1	183	90	64.92
1	172	82	51.42
1	156	45.3	21.02
1	161	59	28.94



- Consider K=5
- Step 4: Test example is assigned the most common class among its K neighbours
  - Class: Adult

Determining K, Number of Neighbours

- This is determined experimentally
- Starting with K=1, test set is used to estimate the accuracy of the classifier
- This process is repeated each time by incrementing K to allow for more neighbour
- The K value that gives the maximum accuracy may be selected
- Preferably the value of K should be an odd number and prime number.

Data Normalization

- Since the distance measure is used, K-NN classifier require normalising the values of each attribute
- Normalising the training data:
  - Compute the minimum and maximum values of each of the attributes in the training data
  - Store the minimum and maximum values of each of the attributes
  - Perform the min-max normalization on training data set
- Normalizing the test data:
  - Use the stored minimum and maximum values of each of the attributes from training set to normalise the test examples
- NOTE: Ensure that test examples are not causing out-of-bound error

Lazy Learning : Learning from Neighbours

- The K nearest neighbour classifier is an example of lazy learner
- Lazy learning waits until the last minute before doing any model construction to classify test example
- When the training examples are given, a lazy learner simply stores them and waits until it is given a test example
- When it sees the test example, then it classify based on its similarity to the stored training examples
- Since the lazy learns stores the training examples or instances, they also called instance based learners
- Disadvantages:
  - Making classification or prediction is computationally intensive
  - Require efficient huge storage techniques when the training samples are huge

Text Books

1. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann Publishers, 2011.

2. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 2009.

3. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

Performance Evaluation for Classification

Confusion Matrix – 2-class

		Actual Class	
		Class1 (Positive)	Class2 (Negative)
Predicted Class	Class1 (Positive)	True Positive (C11)	False Positive (C12)
	Class2 (Negative)	False Negative (C21)	True Negative (C22)

- True Positive: Number of test samples correctly predicted as positive class (Class1).
- True Negative: Number of test samples correctly predicted as negative class (Class2).
- False Positive: Number of test samples predicted as positive class (Class1) but actually belonging to negative class (Class2).
- False Negative: Number of test samples predicted as negative class (Class2) but actually belonging to positive class (Class1).

Confusion Matrix – 2-class

		Actual Class	
		Class1 (Positive)	Class2 (Negative)
Predicted Class	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

Total test samples in class1

Confusion Matrix – 2-class

		Actual Class	
		Class1 (Positive)	Class2 (Negative)
Predicted Class	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

Total test samples in class2

Confusion Matrix – 2-class

		Actual Class	
		Class1 (Positive)	Class2 (Negative)
Predicted Class	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

Total test samples predicted as class1

- Biometric authentication system to access account
  - False Positive (wrongly detecting as genuine person) should be low
  - Some False Negative (Not detecting a genuine person as genuine) is OK
  - Precision should be high

Confusion Matrix – 2-class

		Actual Class	
		Class1 (Positive)	Class2 (Negative)
Predicted Class	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

Total test samples predicted as class2

- Medical image analysis of microscopic image to detect the presence of cancer
  - False Negative (Detecting cancerous image as not cancer) should be low
  - Some False Positive (Detecting a non-cancerous images as cancer) is OK
  - Recall should be high

Accuracy – 2-class

$$\text{Accuracy}(\%) = \frac{\text{Number of samples correctly classified (C11 + C22)}}{\text{Total number of samples used for testing}} * 100$$

$$\text{Accuracy}(\%) = \frac{TP + TN}{\text{Total number of samples used for testing}} * 100$$

		Actual Class	
		Class1 (Positive)	Class2 (Negative)
Predicted Class	Class1 (Positive)	True Positive (C11)	False Positive (C12)
	Class2 (Negative)	False Negative (C21)	True Negative (C22)

Confusion Matrix - Multiclass

Illustration: Number of classes is 3. It can be extended to any number of classes

		Actual Class		
		Class1	Class2	Class3
Predicted Class	Class1	C11	C12	C13
	Class2	C21	C22	C23
	Class3	C31	C32	C33

- C11: Number of test examples predicted as class1 and actually belonging to class1
- C12: Number of test examples predicted as class1, but actually belonging to class2
- C13: Number of test examples predicted as class1, but actually belonging to class3
- *Similarly C21, C22, C23, C31, C32 and C33 are interpreted*

Confusion Matrix - Multiclass

With reference to Class1:

		Actual Class		
		Class1	Class2	Class3
Predicted Class	Class1	C11	C12	C13
	Class2	C21	C22	C23
	Class3	C31	C32	C33

- True Positive: Number of test samples correctly predicted as positive class (class1) (C11).
- True Negative: Number of test samples correctly predicted as negative class (class2 and class3) (C22+C33).
- False Positive: Number of test samples predicted as positive class (class1) but actually belonging to negative class (class2 and class3) (C12+C13)
- False Negative: Number of test samples predicted as negative class (class2 and class3) but actually belonging to positive class (class1) (C21+C31)

Confusion Matrix - Multiclass

With reference to Class2:

		Actual Class		
		Class1	Class2	Class3
Predicted Class	Class1	C11	C12	C13
	Class2	C21	C22	C23
	Class3	C31	C32	C33

- True Positive: Number of test samples correctly predicted as positive class (class2) (C22).
- True Negative: Number of test samples correctly predicted as negative class (class1 and class3) (C11+C33).
- False Positive: Number of test samples predicted as positive class (class2) but actually belonging to negative class (class1 and class3) (C21+C23)
- False Negative: Number of test samples predicted as negative class (class1 and class3) but actually belonging to positive class (class2) (C12+C32)

Confusion Matrix - Multiclass

With reference to Class3:

		Actual Class		
		Class1	Class2	Class3
Predicted Class	Class1	C11	C12	C13
	Class2	C21	C22	C23
	Class3	C31	C32	C33

- True Positive: Number of test samples correctly predicted as positive class (class3) (C33).
- True Negative: Number of test samples correctly predicted as negative class (class1 and class2) (C11+C22).
- False Positive: Number of test samples predicted as positive class (class3) but actually belonging to negative class (class1 and class2) (C31+C32)
- False Negative: Number of test samples predicted as negative class (class1 and class2) but actually belonging to positive class (class3) (C13+C23)

Confusion Matrix - Multiclass

Example: Number of classes = 3. Same concept can be extended to number of classes more than 3

		Actual Class			
		Class1	Class2	Class3	
Predicted Class	Class1	C11	C21	C31	Total samples predicted as class1
	Class2	C12	C22	C32	Total samples predicted as class2
	Class2	C13	C23	C33	Total samples predicted as class3
Total		Total samples in class1	Total samples in class2	Total samples in class3	

Total samples used for testing

Accuracy of Multiclass Classification

Example: Number of classes = 3. Same concept can be extended to number of classes more than 3

$$\text{Accuracy}(\%) = \frac{\text{Number of samples correctly classified } (C11 + C22 + C33)}{\text{Total number of samples used for testing}} * 100$$

$$\text{Accuracy}(\%) = \frac{TP + TN}{\text{Total number of samples used for testing}} * 100$$

		Actual Class		
		Class1	Class2	Class3
Predicted Class	Class1	C11	C21	C31
	Class2	C12	C22	C32
	Class2	C13	C23	C33

Binary (2-class) Classification: Precision, Recall and F-measure

		Actual Class	
		Class1 (Positive)	Class2 (Negative)
Predicted Class	Class1 (Positive)	True Positive (TP)	False Positive (FP)
	Class2 (Negative)	False Negative (FN)	True Negative (TN)

- Precision:
  - Number of samples correctly classified as positive class, out of all the examples classified as positive class
  - It is also called positive predictive value

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Precision} = \frac{\text{Number of samples correctly classified as positive class}}{\text{Total number of samples classified as positive class}}$$

Binary (2-class) Classification: Precision, Recall and F-measure

		Actual Class	
		Class1 (Positive)	Class2 (Negative)
Predicted Class	Class1 (Positive)	True Positive (TP)	False Positive (FP)
	Class2 (Negative)	False Negative (FN)	True Negative (TN)

- Recall:
  - Number of samples correctly classified as positive class, out of all the examples belonging to positive class
  - It is also called as sensitivity or true positive rate (TPR)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{Number of samples correctly classified as positive class}}{\text{Total number of samples belonging to positive class}}$$

Binary (2-class) Classification: Precision, Recall and F-measure

		Actual Class	
		Class1 (Positive)	Class2 (Negative)
Predicted Class	Class1 (Positive)	True Positive (TP)	False Positive (FP)
	Class2 (Negative)	False Negative (FN)	True Negative (TN)

- F-measure or F-score or F1-score:
  - Combines precision and recall
  - Recall and precision are evenly weighted.
  - Harmonic mean of precision and recall

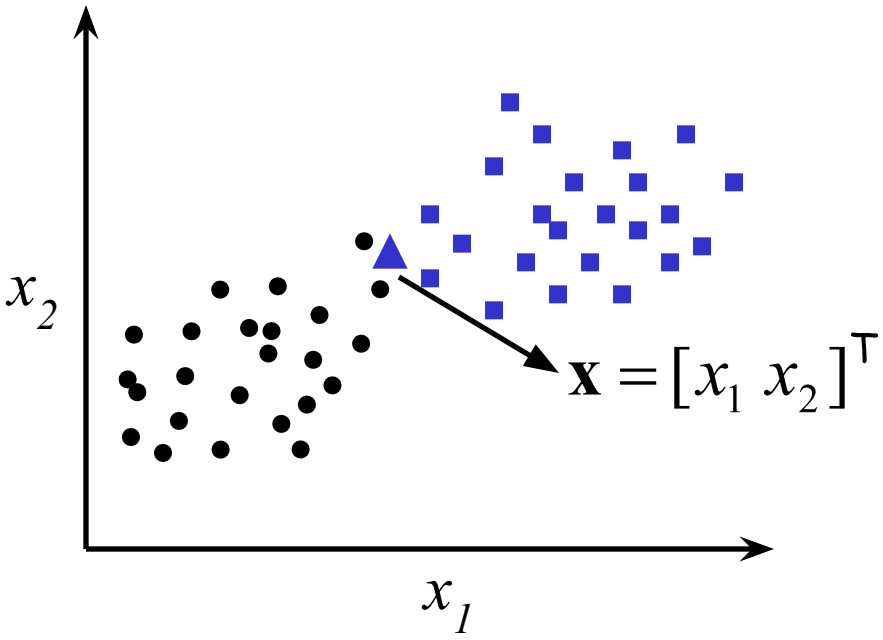
$$\text{F-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Supervised Machine Learning: Pattern Classification

K-Nearest Neighbor, Reference Template Method

K-Nearest Neighbours (K-NN) Method

- Consider the class labels of the K training examples nearest to the test example
- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$



- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Choose the first K examples in the sorted list
  - K is the number of neighbours for text example
- Step 4: Test example is assigned the most common class among its K neighbours

Reference Templates Method

- Each class is represented by its reference templates
  - Mean of each data points of each class as reference template
  - Let the data of class i be $\mathsf{D}_i = \{\mathbf{x}_n\}_{n=1}^{N_i}, \mathbf{x}_n \in \mathsf{R}^d$
    - N_i: Number of examples (data points) in class i
  - Mean of data points of a class i, $\boldsymbol{\mu}_i$ is given as:

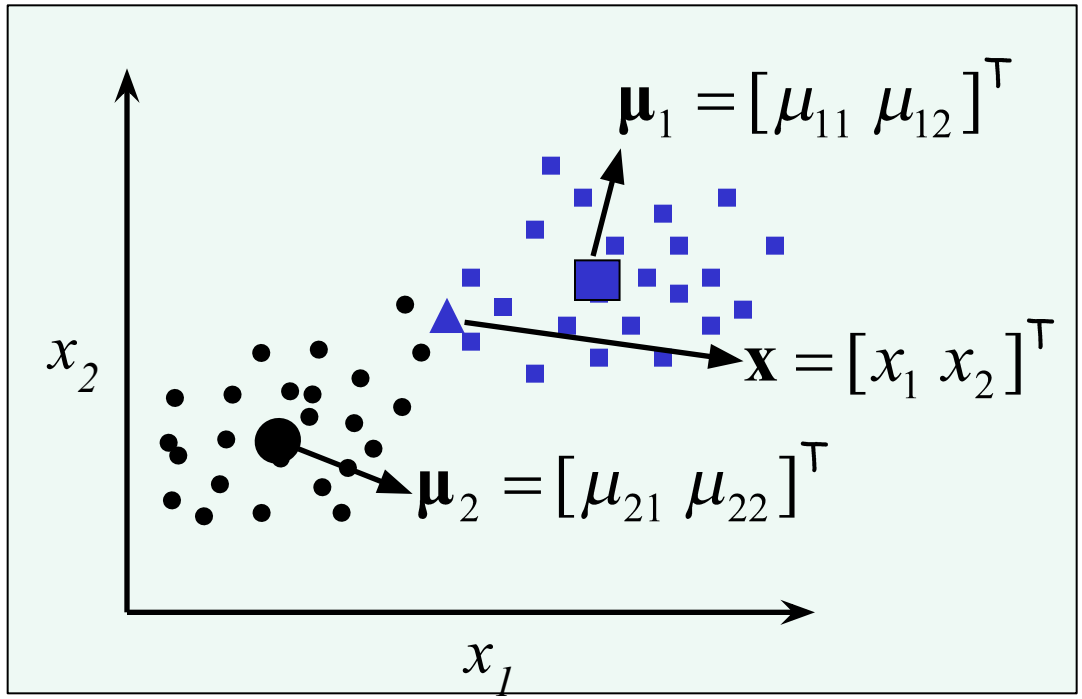
$$\boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{n=1}^{N_i} \mathbf{x}_n$$

Reference Templates Method

- Each class is represented by its reference templates
  - Mean of each data points of each class as reference template
- For a test example, compute an Euclidean distance to all the reference template corresponding to each class, $ED(\mathbf{x}, \boldsymbol{\mu}_i)$

$$= \text{argmin } ED(\mathbf{x}, \boldsymbol{\mu}_i)$$

$\boldsymbol{\mu}_i$: Mean vector of class i



- The class of the nearest reference template (mean) is assigned to the test pattern

Class label for $\mathbf{x} = \underset{i}{\text{argmin}}\ ED(\mathbf{x}, \boldsymbol{\mu}_i)$

$$i = 1,2,\ldots, M$$

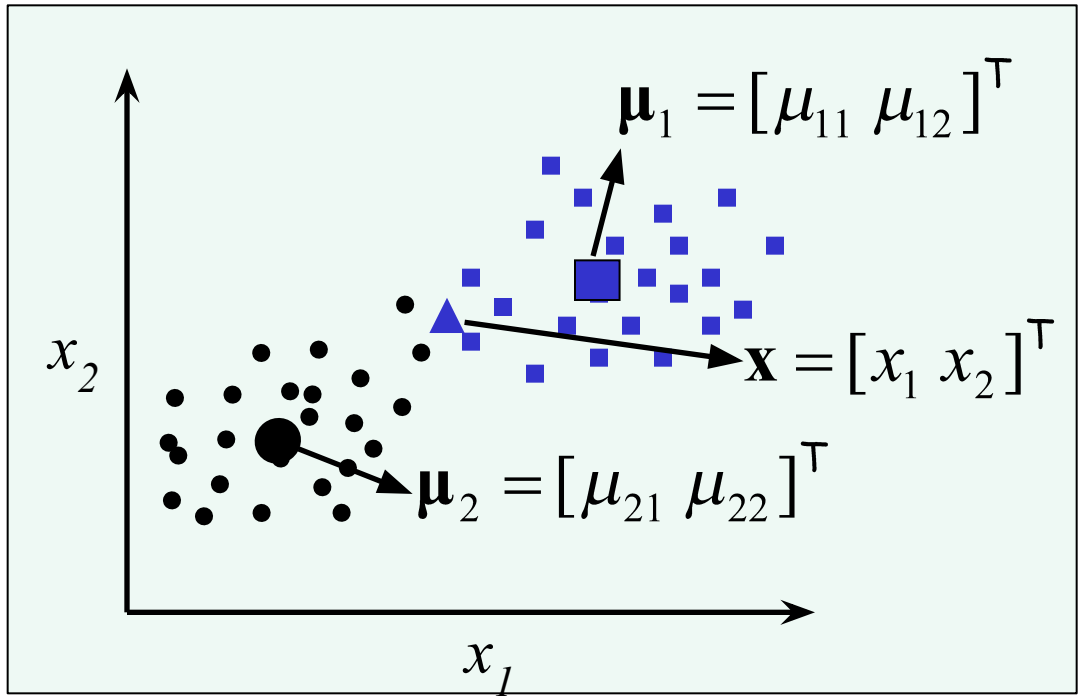
M = Number of classes

Reference Templates Method

- Each class is represented by its reference templates
  - Mean of each data points of each class as reference template
- For a test example, compute an Euclidean distance to all the reference template corresponding to each class, $ED(\mathbf{x}, \boldsymbol{\mu}_i)$

$$= \text{argmin } ED(\mathbf{x}, \boldsymbol{\mu}_i)$$

$\boldsymbol{\mu}_i$: Mean vector of class i



- The class of the nearest reference template (mean) is assigned to the test pattern
- Learning: Estimating first order statistics (mean) from the data of each class

Illustration of Reference Templates Method: Adult(1)-Child(0) Classification

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Training Phase:
  - Compute sample mean vector from training data of class 0 (Child)

$$\boldsymbol{\mu}_0 = [103.60 \quad 30.66]$$

Illustration of Reference Templates Method: Adult(1)-Child(0) Classification

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Training Phase:
  - Compute sample mean vector from training data of class 0 (Child)

$$\boldsymbol{\mu}_0 = [103.60 \;\; 30.66]$$

  - Compute sample mean vector from training data of class 1 (Adult)

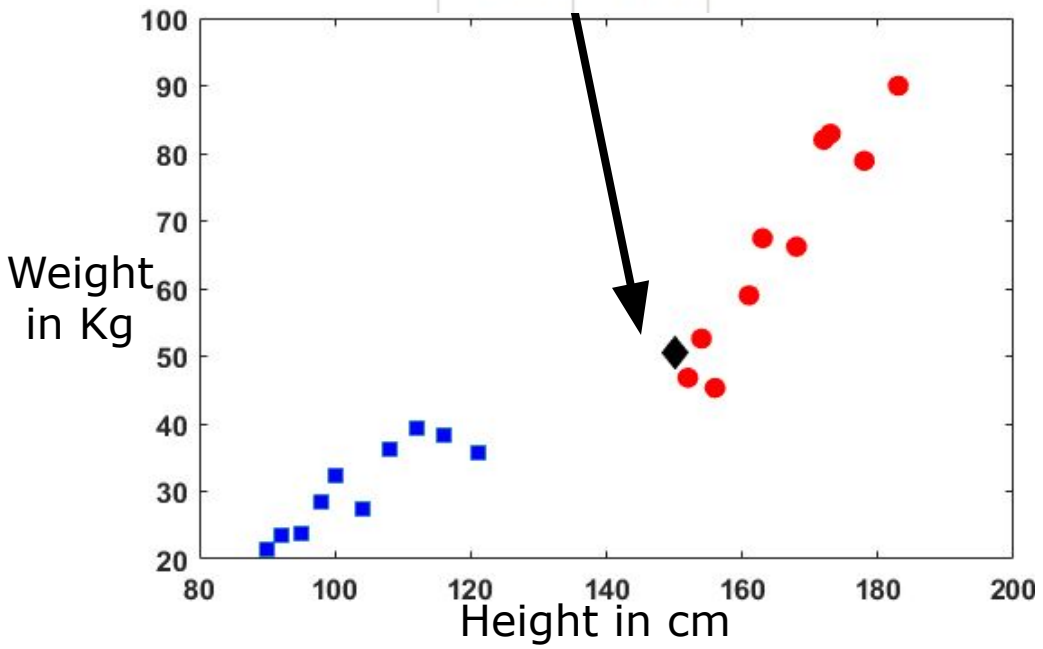
$$\boldsymbol{\mu}_1 = [166.00 \;\; 67.12]$$

Illustration of Reference Templates Method: Adult(1)-Child(0) Classification

- Test Phase - Classification:

	Height	Weight	Class
$\boldsymbol{\mu}_0$	103.60	30.66	0
$\boldsymbol{\mu}_1$	166.60	67.12	1

Test Example, $\mathbf{x}$: 150 50.6



- Compute Euclidean distance of test sample, $\mathbf{x}$ with mean vector of class 0 (Child), $\boldsymbol{\mu}_0$: $ED(\mathbf{x}, \boldsymbol{\mu}_0) = 50.50$
- Compute Euclidean distance of test sample, $\mathbf{x}$ with mean vector of class 1 (Adult), $\boldsymbol{\mu}_1$: $ED(\mathbf{x}, \boldsymbol{\mu}_1) = 23.00$

Class label of $\mathbf{x}$ = Adult

Modified Reference Templates Method

- Each class is represented by its reference templates
  - Mean and variance (covariance) of data points of each class as reference template
  - Let the data of class i be $\mathsf{D}_i = \{\mathbf{x}_n\}_{n=1}^{N_i}, \mathbf{x}_n \in \mathsf{R}^d$
    - N_i: Number of examples (data points) in class i
  - Mean of data points of a class i, $\boldsymbol{\mu}_i$ is given as:

$$\boldsymbol{\mu}_i = \frac{1}{N_i}\sum_{n=1}^{N_i}\mathbf{x}_n$$

  - Covariance matrix of data points of a class i, $\boldsymbol{\Sigma}_i$ is given as:

$$\Sigma_i = \frac{1}{N_i - 1}\sum_{n=1}^{N_i}(\mathbf{x}_n - \boldsymbol{\mu}_i)(\mathbf{x}_n - \boldsymbol{\mu}_i)^{\mathsf{T}}$$

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \ldots \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 \ldots \sigma_{2d} \\ \ldots\ldots \\ \sigma_{d1} & \sigma_{d2} \ldots \sigma_d^2 \end{bmatrix}$$

σ_j^2 is variance; $\sigma_j^2 = \frac{1}{N_i - 1}\sum_{n=1}^{N_i}(x_{nj} - \mu_{ji})^2$

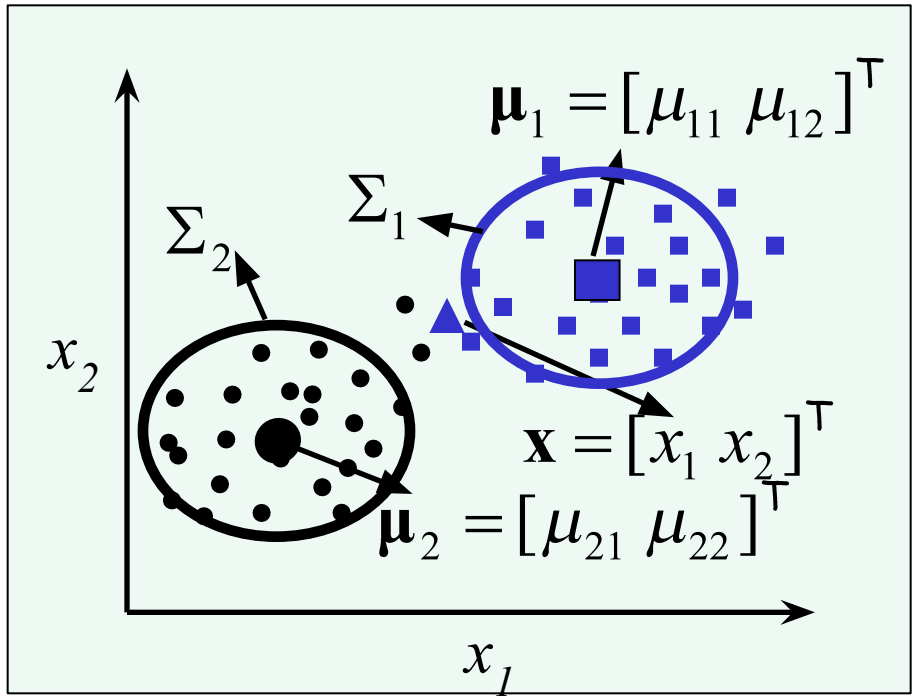
σ_{jk}: Covariance of j^{th} and k^{th} attribute $\sigma_{jk} = \frac{1}{N_i - 1}\sum_{n=1}^{N_i}(x_{nj} - \mu_{ij})(x_{nk} - \mu_{ik})$

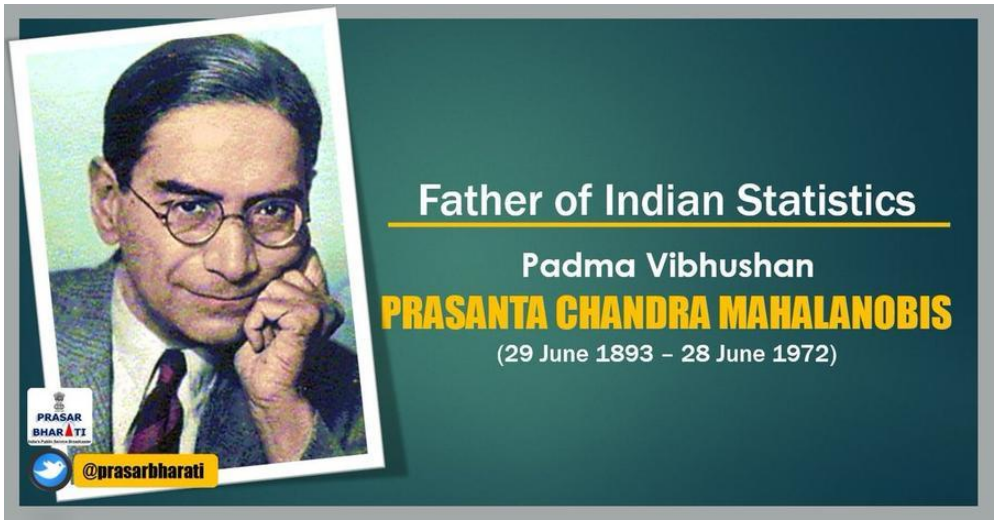
Modified Reference Templates Method

- Each class is represented by one or more reference templates
  - Mean and variance (covariance) of data points of each class as reference template
- For a test example, compute a **Mahalanobis distance** to all the reference template corresponding to each class, $MD(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

$$= \text{argmin } MD(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$\boldsymbol{\mu}_i$ & $\boldsymbol{\Sigma}_i$: Mean vector and Covariance matrix of class i





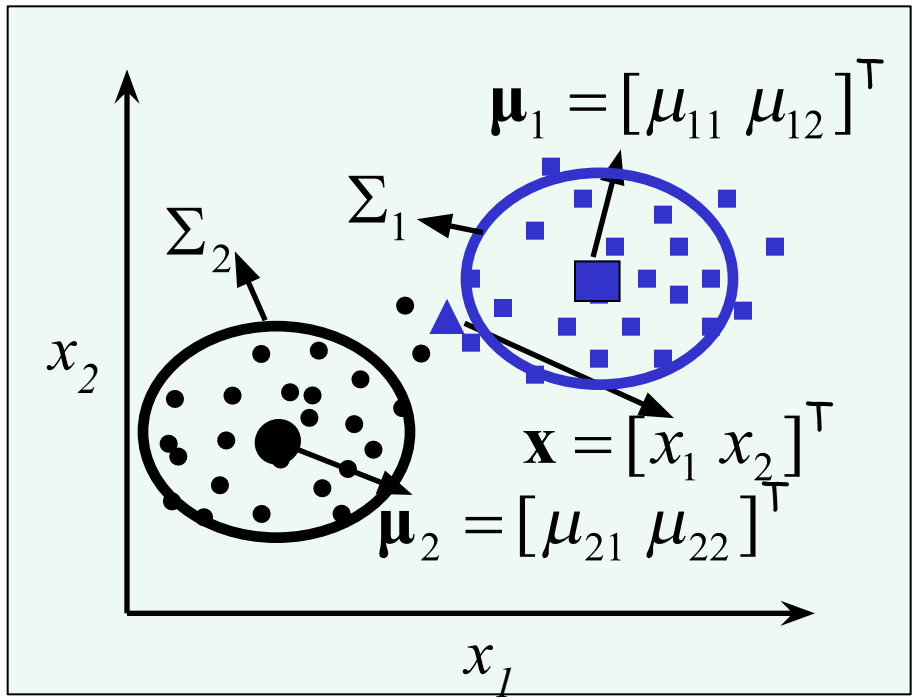
The **Mahalanobis distance** is a measure of the distance between a point and a distribution

Modified Reference Templates Method

- Each class is represented by one or more reference templates
  - Mean and variance (covariance) of data points of each class as reference template
- For a test example, compute a **Mahalanobis distance** to all the reference template corresponding to each class, $MD(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

$$= \text{argmin}\ MD(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \text{argmin}\ \sqrt{(\mathrm{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathrm{x} - \boldsymbol{\mu}_i)}$$

$\boldsymbol{\mu}_i$ & $\boldsymbol{\Sigma}_i$: Mean vector and Covariance matrix of class i



- The class of the nearest reference templates is assigned to the test pattern

Class label for $\mathbf{x} = \underset{i}{\text{argmin}}\ MD(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

$$i = 1,2,\ldots, M$$

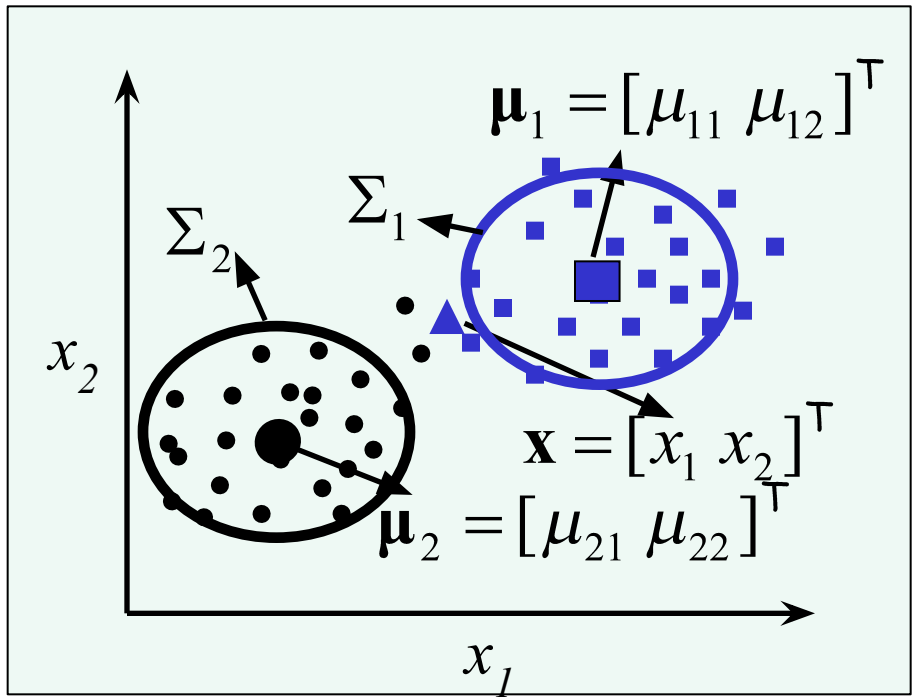
M = Number of classes

Modified Reference Templates Method

- Each class is represented by one or more reference templates
  - Mean and variance (covariance) of data points of each class as reference template
- For a test example, compute a **Mahalanobis distance** to all the reference template corresponding to each class, $MD(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

$$= \text{argmin } MD(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad = \text{argmin } \sqrt{(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)}$$

$\boldsymbol{\mu}_i$ & $\boldsymbol{\Sigma}_i$: Mean vector and Covariance matrix of class i



- The class of the nearest reference templates is assigned to the test pattern
- Learning: Estimating
  - first order statistics (mean) and
  - Second order statistics (variance and covariance) from the data of each class

Illustration of Reference Templates Method: Adult(1)-Child(0) Classification

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Training Phase:
  - Compute sample mean vector from training data of class 0 (Child)

$$\boldsymbol{\mu}_0 = [103.60 \quad 30.66]$$

  - Compute sample covariance matrix from training data of class 0 (Child)

$$\boldsymbol{\Sigma}_0 = \begin{pmatrix} 109.38 & 61.35 \\ 61.35 & 43.54 \end{pmatrix}$$

Illustration of Reference Templates Method: Adult(1)-Child(0) Classification

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Training Phase:
  - Compute sample mean vector from training data of class 0 (Child)

$$\boldsymbol{\mu}_0 = [103.60 \quad 30.66]$$

  - Compute sample covariance matrix from training data of class 0 (Child)

$$\boldsymbol{\Sigma}_0 = \begin{pmatrix} 109.38 & 61.35 \\ 61.35 & 43.54 \end{pmatrix}$$

  - Compute sample mean vector from training data of class 1 (Adult)

$$\boldsymbol{\mu}_1 = [166.00 \quad 67.12]$$

  - Compute sample covariance matrix from training data of class 1 (Adult)

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 110.67 & 160.53 \\ 160.53 & 255.49 \end{pmatrix}$$

Illustration of Reference Templates Method: Adult(1)-Child(0) Classification

- Test Phase - Classification:

$\boldsymbol{\mu}_0 = [103.60 \quad 30.66]$

$\boldsymbol{\Sigma}_0 = \begin{pmatrix} 109.38 & 61.35 \\ 61.35 & 43.54 \end{pmatrix}$

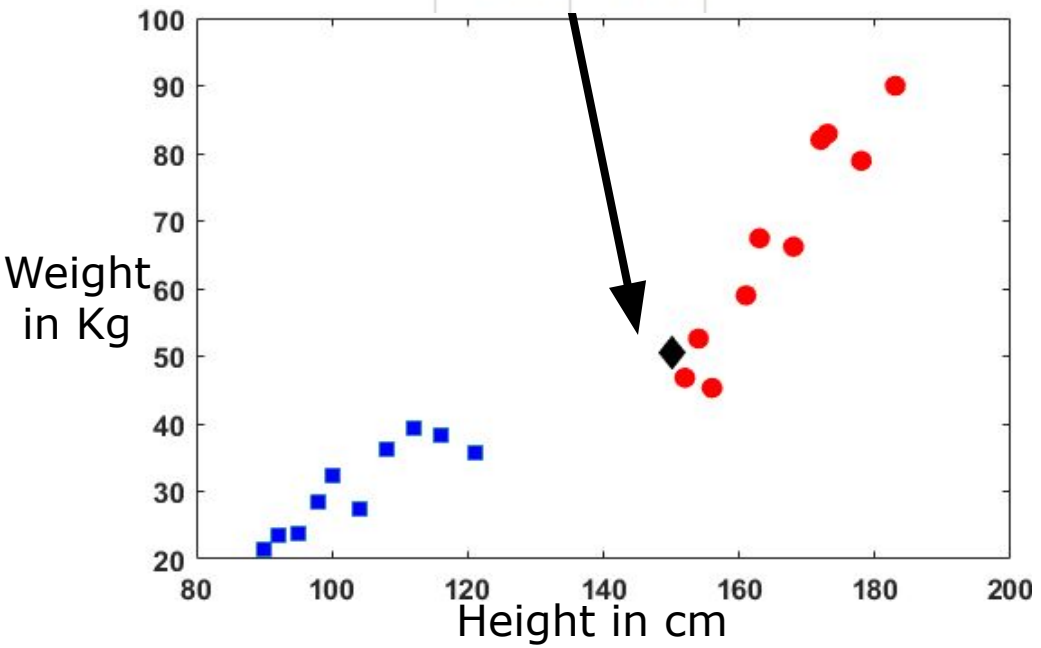
Class
0

$\boldsymbol{\mu}_1 = [166.00 \quad 67.12]$

$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 110.67 & 160.53 \\ 160.53 & 255.49 \end{pmatrix}$

Class
1

Test Example, $\mathbf{x}$: 150 50.6



- Compute Mahalanobis distance of test sample, $\mathbf{x}$ with mean vector and covariance matrix of class 0 (Child): $MD(\mathbf{x}, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) = 4.87$
- Compute Mahalanobis distance of test sample, $\mathbf{x}$ with mean vector and covariance matrix of class 1 (Adult): $MD(\mathbf{x}, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) = 2.07$

Class label of $\mathbf{x}$ = Adult

Classification using Reference Template Methods

- For a test example, a distance measure is computed with the reference template of each class
- The class of the reference template with least distance is assigned to the test pattern
- When Mahalanobis distance is used, it gives the notion that distance measure is computed between a test example and the distribution (density) of a class
  - Distribution (density) of class: All the training examples are drawn from that distribution
  - Density here is normal (Gaussian) density
- In other way, we are interested to estimate probability of class, $P(C_i \mid \mathbf{x})$
  - Given the test example $\mathbf{x}$, what is the probability that it belongs to i^{th} class (C_i)
- Solution: Bayes classifier

Text Books

1. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann Publishers, 2011.

2. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 2009.

3. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.