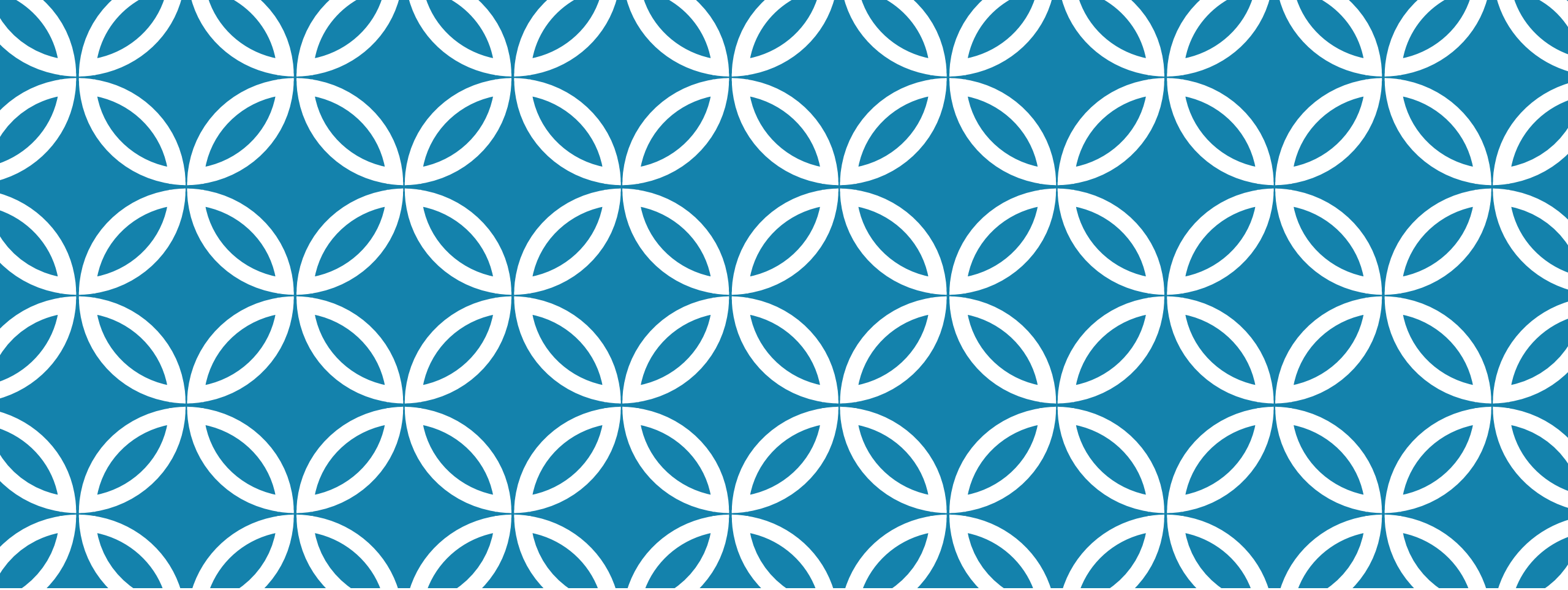


DATA MINING – UNIT 1

By Rashmi Bhattad

WHAT IS DATA MINING?

- Automatically discovering useful information
- Novel and useful patterns
- Different than Information Retrieval
- Data Mining and Knowledge Discovery (KDD)
- Pre-processing
- Closing-the-loop (DM Results → Decision Support System)
- Post-processing

DATA MINING: DEFINITIONS

- *Data mining or knowledge in databases, as it is also known, is the non-trivial extraction of implicit, previously unknown and potentially useful information from the data. This encompasses a number of technical approaches, such as clustering, data summarization, classification, finding dependency networks, analysing changes, and detecting anomalies.*

DATA MINING: DEFINITIONS

- *Data mining is the search for relationships and global partners that exists in large databases but are hidden among vast amounts of data, such as relationship between patient data and their medical diagnosis. This relationship represents valuable knowledge about the database, and the objects in the database, if the database is the faithful mirror of the real world registered by the database.*

DATA MINING: DEFINITIONS

- *Discovering relations that connect variables in a database is a subject of data mining. A data mining system self learn from the previous history of the investigated system, formulating and testing hypothesis about rules which systems obey. When concise and valuable knowledge about the system of interest is discovered, it can and should be interpreted into some decision support system, which helps the manager to make wise and informed business decision.*

DATA MINING: DEFINITIONS

- *Data mining is the process of discovering meaningful, new corelation patterns and trends by sifting through large amount of data stored in repositories, using pattern recognition techniques as well as statistical and mathematical techniques,*

KNOWLEDGE DISCOVERY IN DATABASE

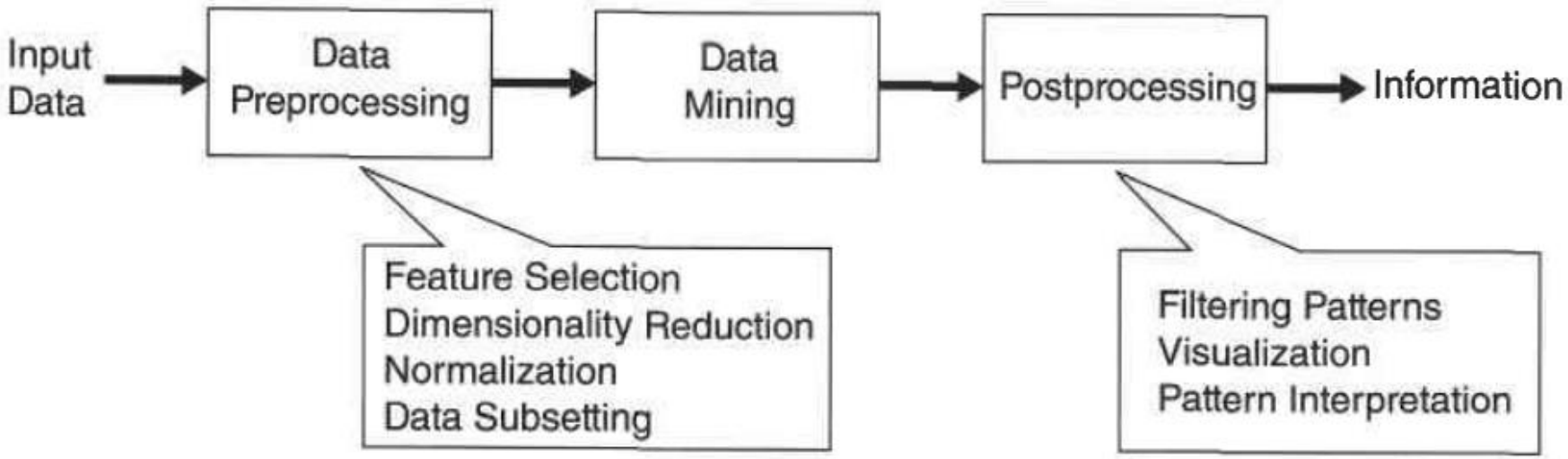


Figure 1.1. The process of knowledge discovery in databases (KDD).

WHAT IS DATA MINING?

- Post-processing:
- Statistical measures
- Hypothesis testing methods
- Spurious data mining results

MOTIVATING CHALLENGES

- Scalability:
  - Novel data structure → efficient retrieval
  - Out-of-edge algorithm → fit in main memory
- High Dimensionality
  - Temporal and Spatial Data
- Heterogeneous and Complex data
  - Web pages, semi-structured text, hyperlinks, 3-d data, climate data, etc
  - Graph connectivity, temporal-spatial autocorrelation, parent child relation
- Data Ownership and Distribution

MOTIVATING CHALLENGES

- Data Ownership and Distribution
  - How to reduce the amount of communication needed to perform the distributed computation
  - How to effectively consolidate data mining results obtained from multiple sources
  - How to address data security issues
- Non-traditional Analysis
  - Generation and evaluation of thousands of hypothesis and,
  - Consequently need of techniques to automate the processes.

THE ORIGINS OF DATA MINING

- Adopt ideas from other areas:
  - Optimization, evolutionary computing, Signal processing, IR, visualization, and information theory.
- Data Mining draws upon ideas such as:
  - Sampling, estimation, hypothesis testing from statistics, and
  - Search Algorithms, modelling techniques, and
  - Learning theories from Artificial Intelligence, pattern recognition, and Machine Learning

THE ORIGINS OF DATA MINING

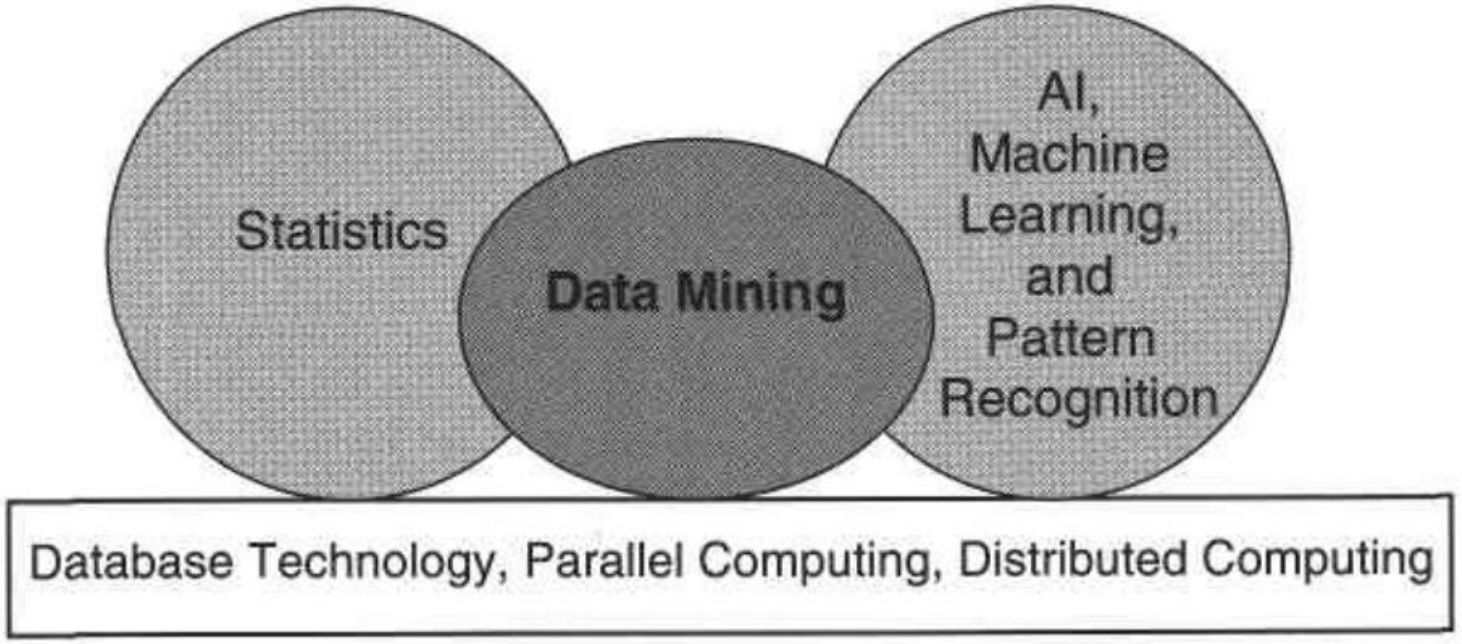


Figure 1.2. Data mining as a confluence of many disciplines.

DATA MINING TASKS:

- Predictive Tasks:
  - Target/Dependent Variable
  - Explanatory/Independent Variable
- Descriptive Tasks:
  - Derives patterns that summarize underlying relationships
  - Correlations, trends, clusters, trajectories, and anomalies
  - Requires postprocessing to validate and explain results

DATA MINING TASKS:

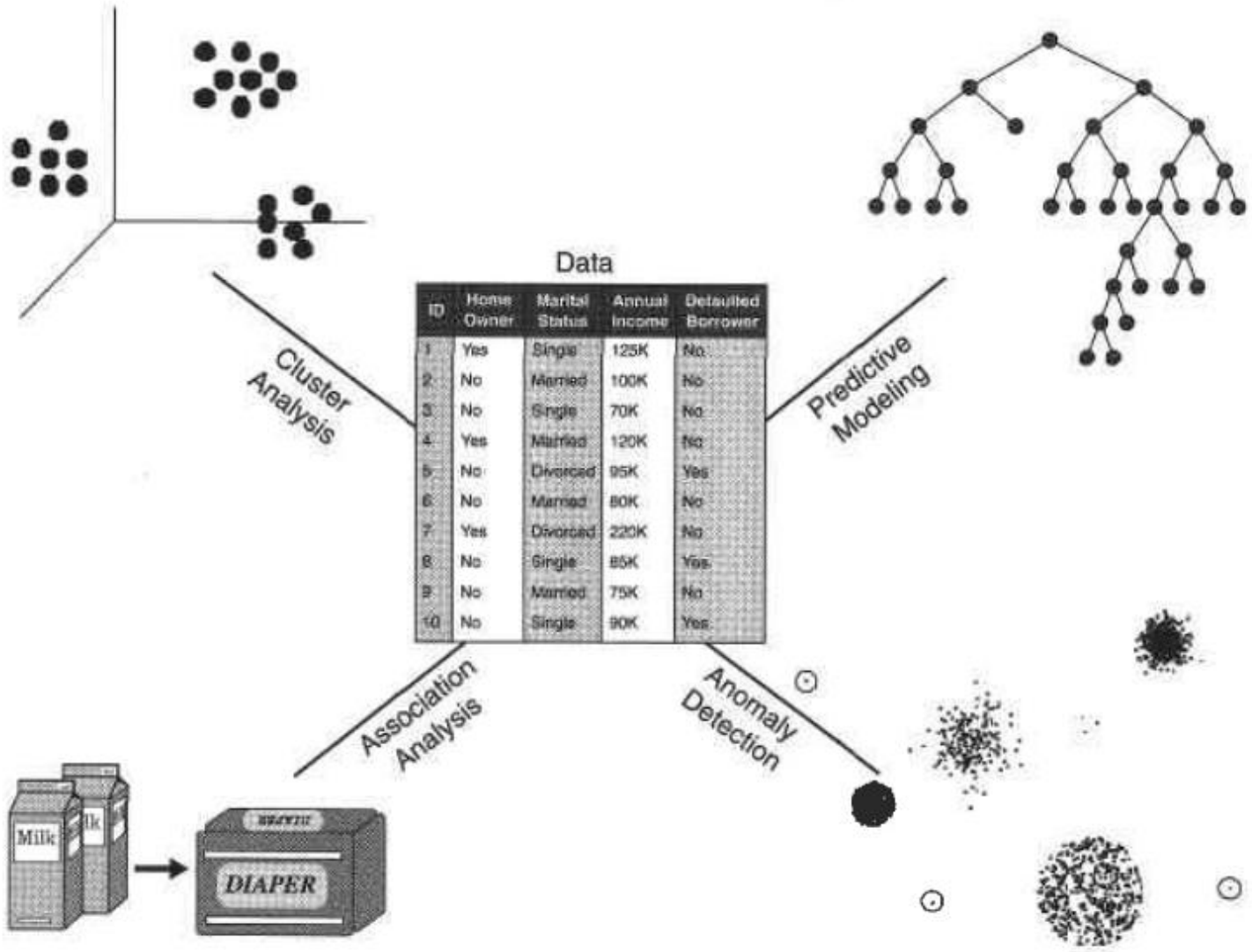


Figure 1.3. Four of the core data mining tasks.

DATA MINING TASKS:

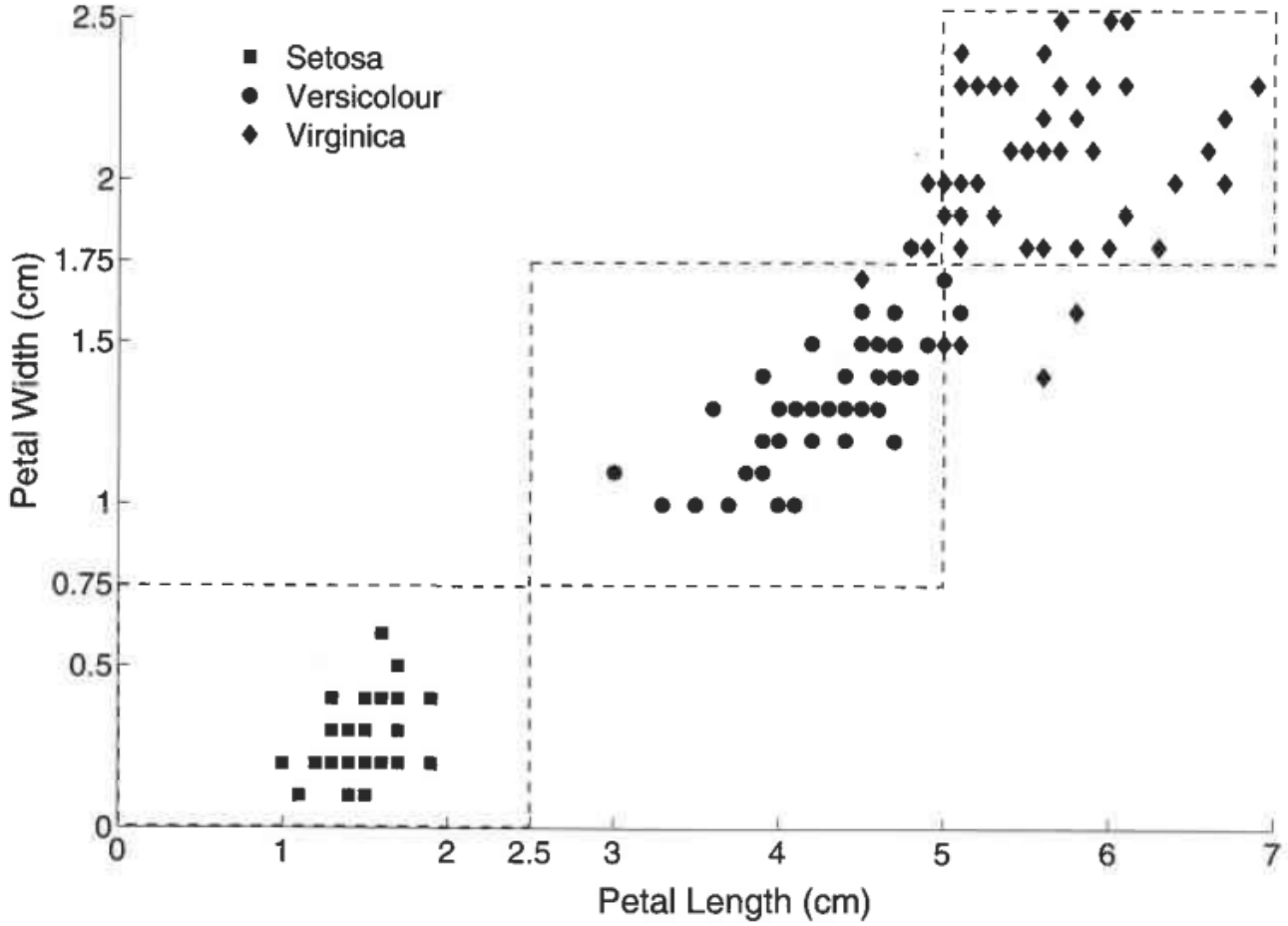


Figure 1.4. Petal width versus petal length for 150 Iris flowers.

DATA MINING TASKS:

- Predictive Tasks: Classification, Regression
- Association Mining: Market Basket Analysis
  - Discovers patterns that strongly associate features in the data
  - Extract the most interesting patterns in an efficient manner
  - Finding groups of genes having related functionality
- Clustering: Document Clustering

ASSOCIATION MINING: MARKET BASKET ANALYSIS

Transaction ID	Items
1	{Bread, Butter, Diapers, Milk}
2	{Coffee, Sugar, Cookies, Salmon}
3	{Bread, Butter, Coffee, Diapers, Milk, Eggs}
4	{Bread, Butter, Salmon, Chicken}
5	{Eggs, Bread, Butter}
6	{Salmon, Diapers, Milk}
7	{Bread, Tea, Sugar, Eggs}
8	{Coffee, Sugar, Chicken, Eggs}
9	{Bread, Diapers, Milk, Salt}
10	{Tea, Eggs, Cookies, Diapers, Milk}

DATA MINING TASKS:

- Anomaly Detection:
  - Outliers
  - High detection rate and false alarm rate
- Credit Card Fraud Detection, network intrusions, unusual patterns of disease, and ecosystem disturbances

DATA:

- Types of Data
  - Qualitative or Quantitative
- Quality of data:
- Pre-processing steps to make data more suitable for data mining
- Analysing data in terms of their relationships

TYPES AND QUALITY OF DATA:

- Attributes used to describe data objects:
  - Qualitative or Quantitative
- Special characteristics:
  - Time series data,
  - Explicit relationship with each other
- Type of data determines tools and techniques to analyze
- Quality of data:
  - Noise, outliers, missing and inconsistent, Biased

TYPES OF DATA: ATTRIBUTES AND MEASUREMENTS

- Data set → Collection of Data Object
  - record, point, vector, entity, pattern, observation, event, sample
- 1. Attributes and Measurements:
  - What is an attribute:
    - attribute: property/char of an object that may vary either from one object to another or from one time to another
    - measurement scale: rule/function that associates a numerical or symbolic value with an attribute of an object
  - Type of an attribute (ID, age)(length of the line segment)
  - Different types of attributes
  - Describing attributes by number of values: Discrete(binary) and Continuous
  - Asymmetric Attributes (presence of non-zero attribute value)

TYPES OF DATA: ATTRIBUTES AND MEASUREMENTS

Attribute Type		Description	Examples	Operations
Categorical (Qualitative)	Nominal	The values of a nominal attribute are just different names; i.e., nominal values provide only enough information to distinguish one object from another. $(=, \neq)$	zip codes, employee ID numbers, eye color, gender	mode, entropy, contingency correlation, χ^2 test
	Ordinal	The values of an ordinal attribute provide enough information to order objects. $(<, >)$	hardness of minerals, $\{good, better, best\}$, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric (Quantitative)	Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, t and F tests
	Ratio	For ratio variables, both differences and ratios are meaningful. $(*, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

TYPES OF DATA: ATTRIBUTES AND MEASUREMENTS

Table 2.3. Transformations that define attribute levels.

Attribute Type		Transformation	Comment
Categorical (Qualitative)	Nominal	Any one-to-one mapping, e.g., a permutation of values	If all employee ID numbers are reassigned, it will not make any difference.
	Ordinal	An order-preserving change of values, i.e., $new\_value = f(old\_value)$, where f is a monotonic function.	An attribute encompassing the notion of good, better, best can be represented equally well by the values $\{1, 2, 3\}$ or by $\{0.5, 1, 10\}$.
Numeric (Quantitative)	Interval	$new\_value = a * old\_value + b$, a and b constants.	The Fahrenheit and Celsius temperature scales differ in the location of their zero value and the size of a degree (unit).
	Ratio	$new\_value = a * old\_value$	Length can be measured in meters or feet.

TYPES OF DATA: TYPES OF DATA SETS

- Types of data sets:
  - General characteristics of the dataset:
    - Dimensionality,
    - Sparsity,
    - Resolution,
  - Record data:
    - Transaction/market basket data,
    - The data matrix
    - The sparse data matrix

DIFFERENT TYPES OF RECORD DATA:

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

(a) Record data.

TID	ITEMS
1	Bread, Soda, Milk
2	Beer, Bread
3	Beer, Soda, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Soda, Diaper, Milk

(b) Transaction data.

Projection of x Load	Projection of y Load	Distance	Load	Thickness
10.23	5.27	15.22	27	1.2
12.65	6.25	16.22	22	1.1
13.54	7.23	17.34	23	1.2
14.27	8.43	18.45	25	0.9

(c) Data matrix.

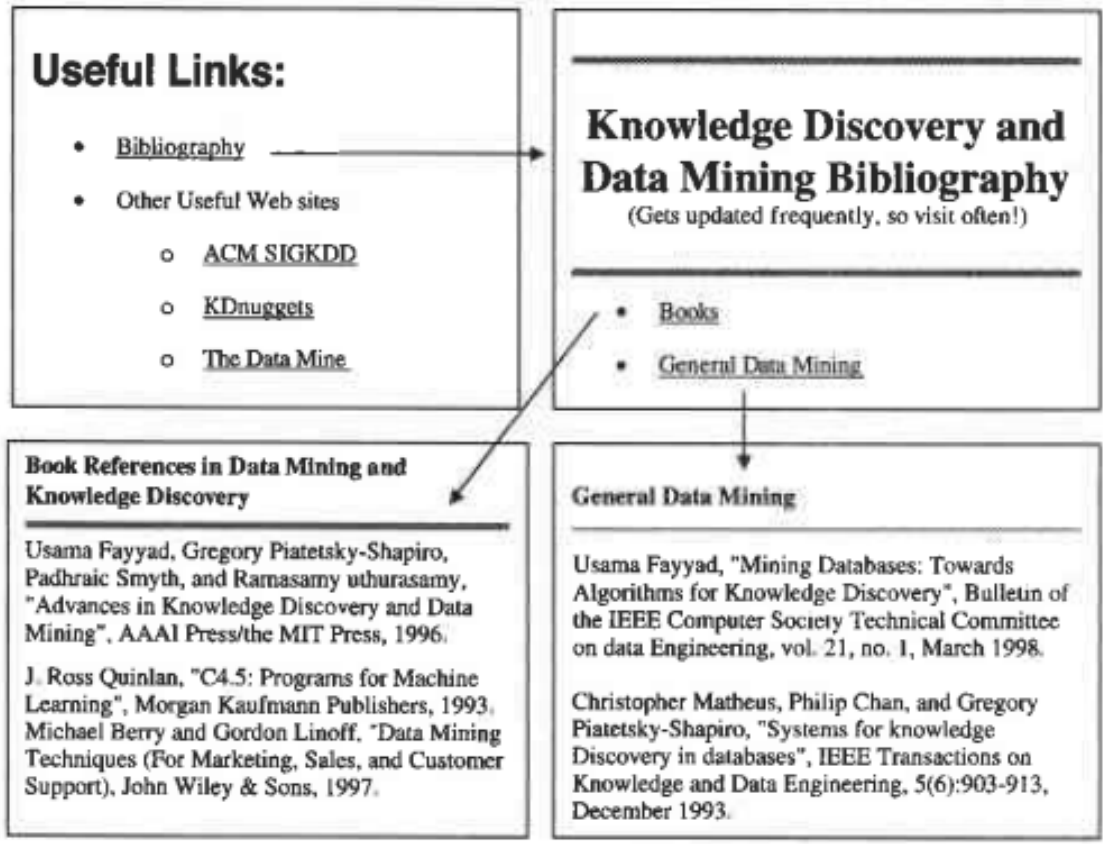
	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

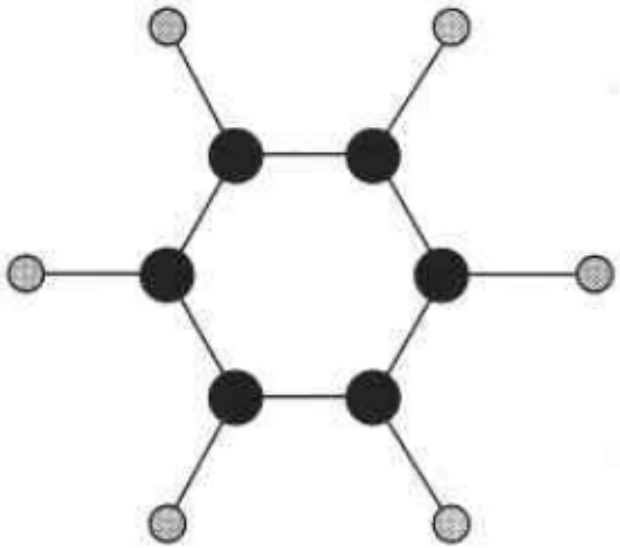
(d) Document-term matrix.

TYPES OF DATA: TYPES OF DATA SETS

- Graph-based data:
  - Data with relationships among objects,
  - Data with objects that are graphs
- Ordered data:
  - Sequential data
  - Sequence data
  - Time series data
  - Spatial data
  - Handling non record data

GRAPH DATA





(a) Linked Web pages.

(b) Benzene molecule.

Figure 2.3. Different variations of graph data.

ORDERED DATA

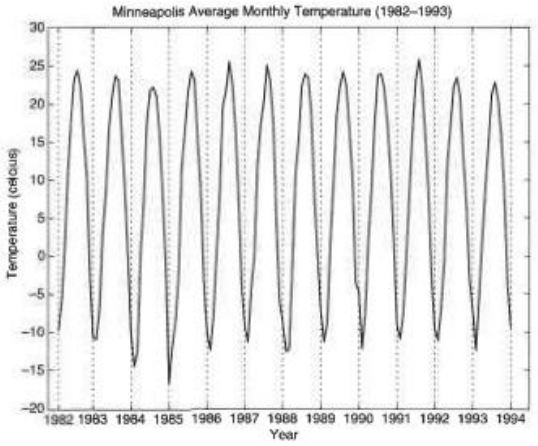
Time	Customer	Items Purchased
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

Customer	Time and Items Purchased
C1	(t1: A,B) (t2:C,D) (t5:A,E)
C2	(t3: A, D) (t4: E)
C3	(t2: A, C)

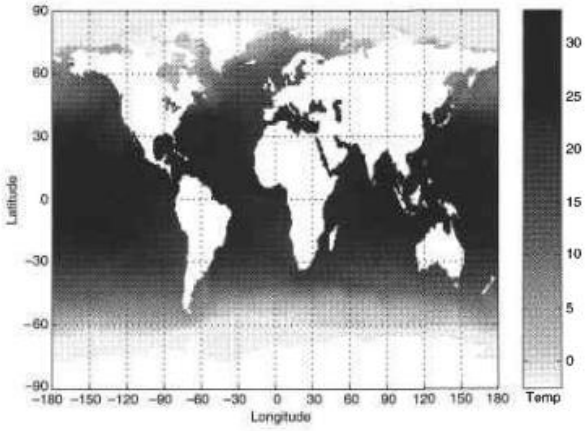
(a) Sequential transaction data.

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

(b) Genomic sequence data.



(c) Temperature time series.



(d) Spatial temperature data.

Figure 2.4. Different variations of ordered data.

DATA PRE-PROCESSING:

- Aggregation
- Sampling
- Dimensionality Reduction
- Feature Subset Selection
- Feature Creation
- Discretization and Binarization
- Variable Transformation

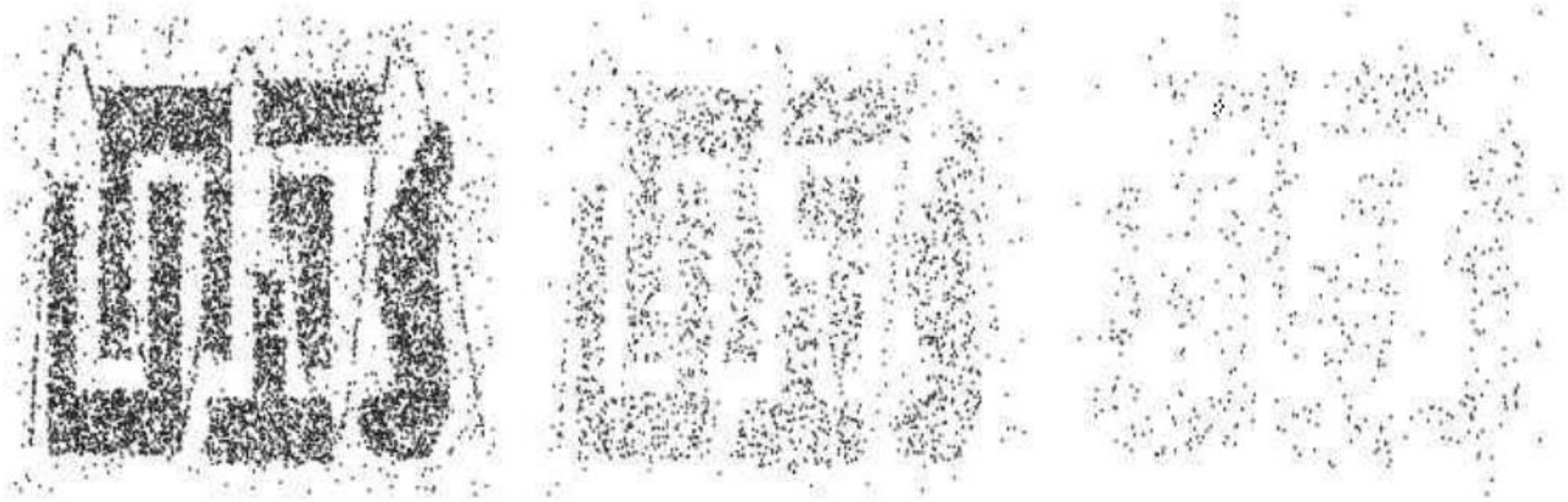
DATA PRE-PROCESSING: AGGREGATION

- OLAP: Online analytical processing
- Smaller dataset: hence memory and processing time
- May use expensive data mining algorithms.
- Change of scope using high level data
- Disadvantage: Potential loss of interesting information

DATA PRE-PROCESSING: SAMPLING

- Sampling in statistics vs Sampling in Data Mining
- Sample should be representative
  - If it posses the same property as original one
- Simple Random Sampling
  - Sampling without replacement
  - Sampling with replacement
- Stratified Sampling
- Prespecified group of objects that too with equal proportions.

DATA PRE-PROCESSING: SAMPLING



(a) 8000 points (b) 2000 points (c) 500 points

Figure 2.9. Example of the loss of structure with sampling.

DATA PRE-PROCESSING: SAMPLING, DR

- Adaptive or Progressive Sampling
- Dimensionality Reduction
  - Curse of dimensionality
  - Principal Component Analysis
    - Linear combination of original attributes
    - Orthogonal(perpendicular) to each other
    - Capture maximum amount of variation in data

DATA PRE-PROCESSING: FEATURE SUBSET SELECTION

- Redundant features, irrelevant features
- Embedded Approaches: Decision Tree
- Filter Approaches: Low pairwise corelation
- Wrapper Approaches: Black box
- Architecture for Feature Subset Selection: evaluating subset, search strategy that control generation of new feature subsets, stopping criteria, validation procedure

DATA PRE-PROCESSING: FEATURE SUBSET SELECTION

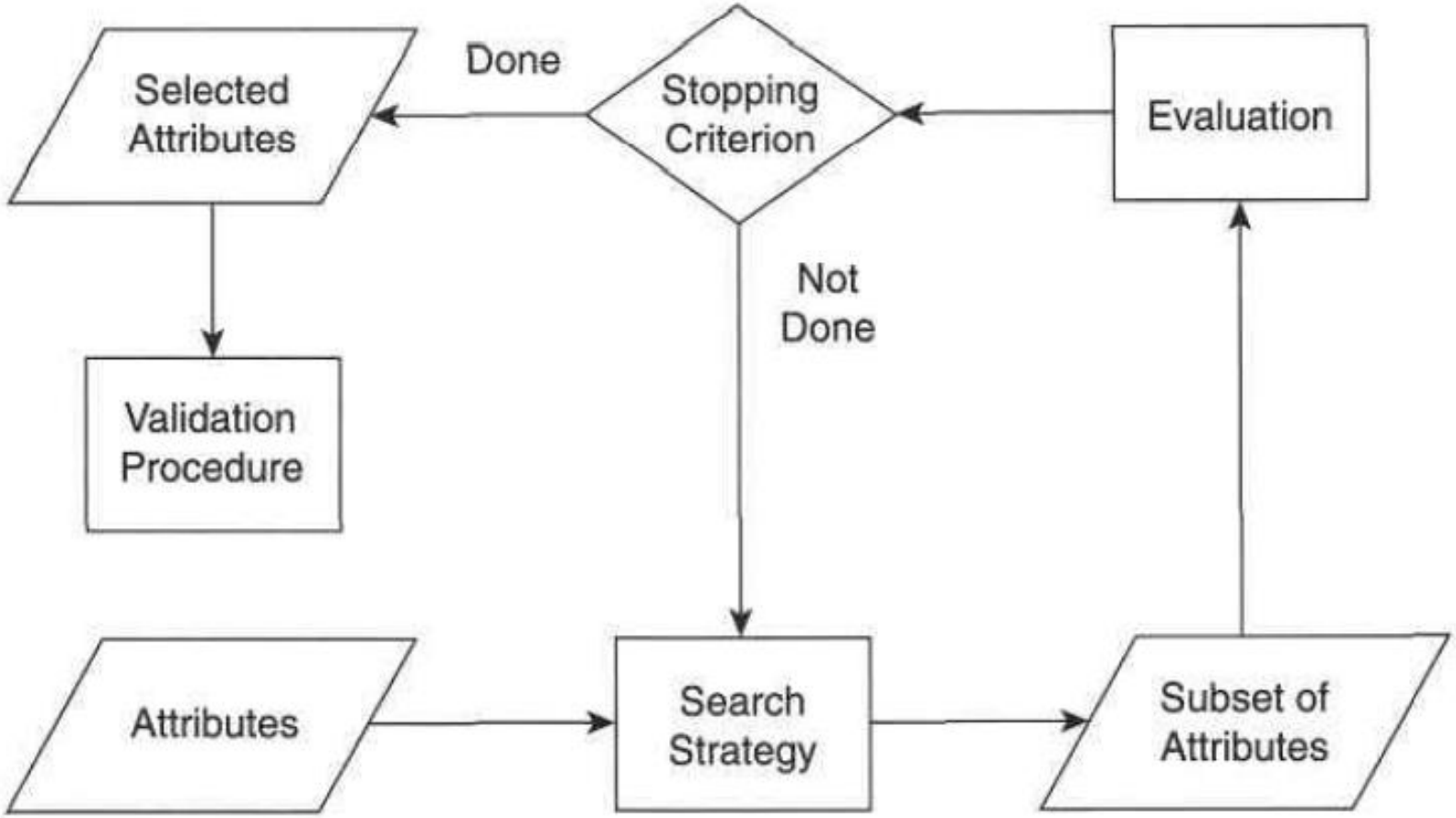


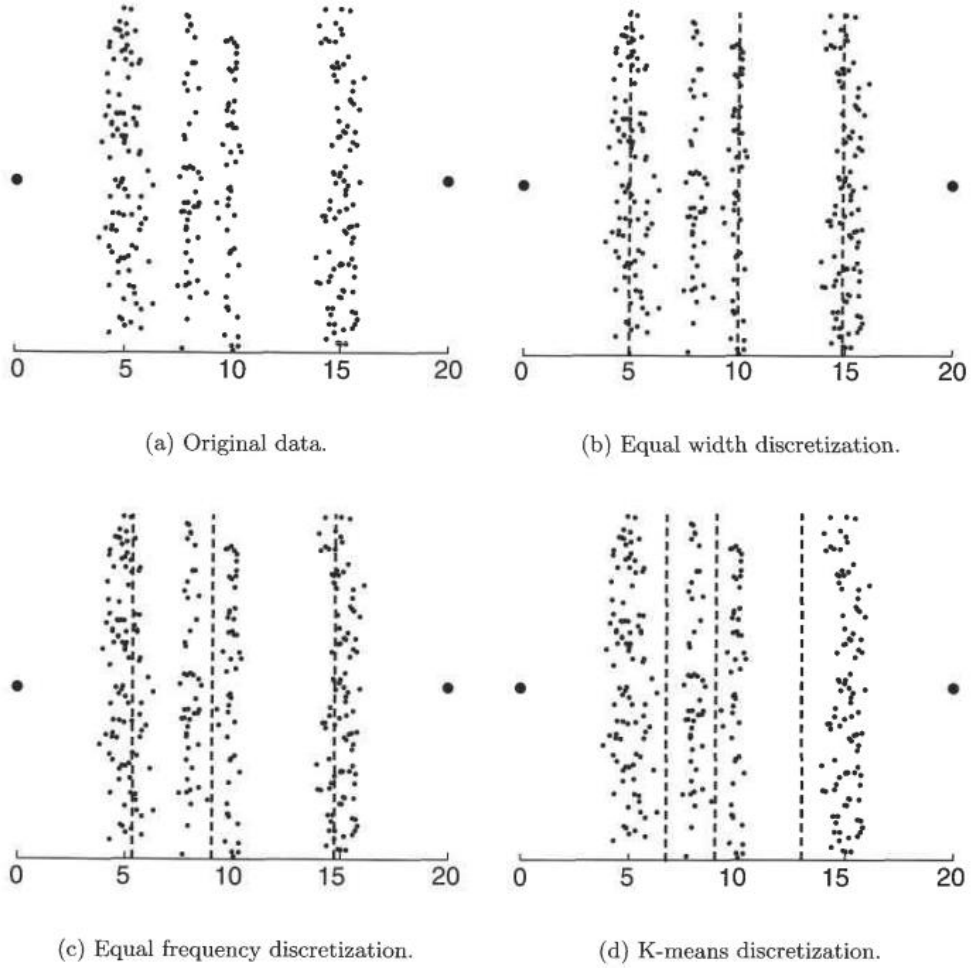
Figure 2.11. Flowchart of a feature subset selection process.

If we are using REINFORCEREINFORCE to learn the parameters, what will be the update for w_1w_1 and w_2w_2, when we pull action a_1

DATA PRE-PROCESSING: FEATURE SUBSET SELECTION

- Feature Weighting
- Feature Creation:
  - Feature Extraction
  - Mapping the data to new space
  - Feature Construction
- Discretization and Binarization
- Variable Transformation

DATA PRE-PROCESSING: FEATURE SUBSET SELECTION



(a) Original data.

(b) Equal width discretization.

(c) Equal frequency discretization.

(d) K-means discretization.

Figure 2.13. Different discretization techniques.

MEASURE OF SIMILARITY AND DISSIMILARITY:

- Basics
- Similarity and Dissimilarity between Simple attributes
- Dissimilarity Between Data objects
- Similarity Between Data Objects
- Example in Proximity Measure
- Issue in Proximity Calculation
- Selecting Right Proximity Measure

MEASURE OF SIMILARITY AND DISSIMILARITY: BASICS

- Used in Clustering, Classification, and Anomaly Detection
- Proximity to measure similarity/dissimilarity
- Similarity
  - Numeric measure of how alike data objects are
  - Is higher when data objects are more alike
  - Often falls in the range [0,1]
- Dissimilarity
  - Numeric measure of how different data objects are
  - Is lower when data objects are more alike
  - Minimum dissimilarity is often 0, the upper limit varies

MEASURE OF SIMILARITY AND DISSIMILARITY: SIMILARITY AND DISSIMILARITY BETWEEN SIMPLE ATTRIBUTES

Table 2.7. Similarity and dissimilarity for simple attributes

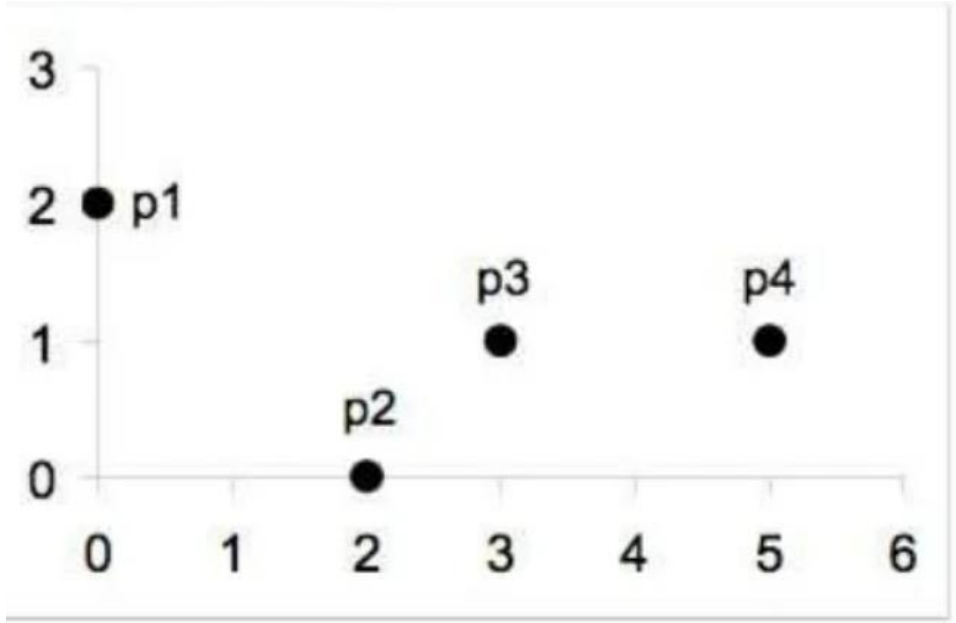
Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \lvert x - y \rvert / (n-1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

MEASURE OF SIMILARITY AND DISSIMILARITY: DISSIMILARITY BETWEEN DATA OBJECTS WITH MULTIPLE ATTRIBUTES

- Dissimilarity Between Data objects with multiple attributes:
- Distance
  - Euclidean Distance: Positivity, Symmetric, Triangular Inequality
  - Minkowski Distance
  - Manhattan Distance
  - Supremum Distance

MEASURE OF SIMILARITY AND DISSIMILARITY:

- Dissimilarity Between Data objects with multiple attributes
-



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

MEASURE OF SIMILARITY AND DISSIMILARITY:

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

MEASURE OF SIMILARITY AND DISSIMILARITY:

Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):

(a) Compute the *Euclidean distance* between the two objects.

(b) Compute the *Manhattan distance* between the two objects.

(c) Compute the *Minkowski distance* between the two objects, using $p = 3$.

MEASURE OF SIMILARITY AND DISSIMILARITY:

Answer:

(a) Compute the *Euclidean distance* between the two objects.

$$\begin{aligned} d(i,j) &= \sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\cdots+(x_{in}-x_{jn})^2} \\ &= \sqrt{|22-20|^2+|1-0|^2+|42-36|^2+|10-8|^2} = 6.71 \end{aligned}$$

(b) Compute the *Manhattan distance* between the two objects.

$$\begin{aligned} d(i,j) &= |x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\cdots+|x_{in}-x_{jn}| \\ &= |22-20|+|1-0|+|42-36|+|10-8| = 11 \end{aligned}$$

(c) Compute the *Minkowski distance* between the two objects, using $p = 3$.

$$\begin{aligned} d(i,j) &= (|x_{i1}-x_{j1}|^q+|x_{i2}-x_{j2}|^q+\cdots+|x_{in}-x_{jn}|^p)^{1/p} \\ &= (|22-20|^3+|1-0|^3+|42-36|^3+|10-8|^3)^{1/3} = 6.15 \end{aligned}$$

MEASURE OF SIMILARITY AND DISSIMILARITY:

- A contingency table for binary data

		Object j		
		1	0	sum
Object i	1	q	r	$q+r$
	0	s	t	$s+t$
	sum	$q+s$	$r+t$	p

- Distance measure for symmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

- Distance measure for asymmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s}$$

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

- Note: Jaccard coefficient is the same as "coherence":

$$coherence(i,j) = \frac{sup(i,j)}{sup(i) + sup(j) - sup(i,j)} = \frac{q}{(q+r)+(q+s)-q}$$

MEASURE OF SIMILARITY AND DISSIMILARITY: SIMILARITY BETWEEN DATA OBJECTS

- Similarity Between Data Objects
  - Symmetry and Positivity hold but triangle inequality does not
- Similarity measure can be converted to the metric distance
- Jaccard and cosine similarity
- Non-symmetric similarity measure

MEASURE OF SIMILARITY AND DISSIMILARITY: EXAMPLE IN PROXIMITY MEASURE

- Similarity Measures for Binary Data
  - Similarity Coefficients
  - Range [0,1]
  - Simple matching coefficient: x, y objects with *n* attributes

$$SMC = \frac{\text{number of matching attribute values}}{\text{number of attributes}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}.$$

MEASURE OF SIMILARITY AND DISSIMILARITY: EXAMPLE IN PROXIMITY MEASURE

- Similarity Measures for Binary Data
  - Simple matching coefficient: x, y objects with *n* attributes

$$SMC = \frac{\text{number of matching attribute values}}{\text{number of attributes}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}.$$

  - Jaccard Coefficient

$$J = \frac{\text{number of matching presences}}{\text{number of attributes not involved in 00 matches}} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}.$$

$$\mathbf{x} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$
$$\mathbf{y} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)$$

MEASURE OF SIMILARITY AND DISSIMILARITY: EXAMPLE IN PROXIMITY MEASURE

- Similarity Measures for Binary Data
  - Cosine Similarity:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|},$$

$$\mathbf{x} = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$$
$$\mathbf{y} = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$$

  - Extended Jaccard Coefficient

$$EJ(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \mathbf{x} \cdot \mathbf{y}}.$$

MEASURE OF SIMILARITY AND DISSIMILARITY: EXAMPLE IN PROXIMITY MEASURE

- Corelation:

$$\text{corr}(\mathbf{x}, \mathbf{y}) = \frac{\text{covariance}(\mathbf{x}, \mathbf{y})}{\text{standard\_deviation}(\mathbf{x}) * \text{standard\_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x \; s_y},$$

$$\text{covariance}(\mathbf{x}, \mathbf{y}) = s_{xy} = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \overline{x})(y_k - \overline{y})$$

$$\text{standard\_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (x_k - \overline{x})^2}$$

$$\text{standard\_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (y_k - \overline{y})^2}$$

MEASURE OF SIMILARITY AND DISSIMILARITY:

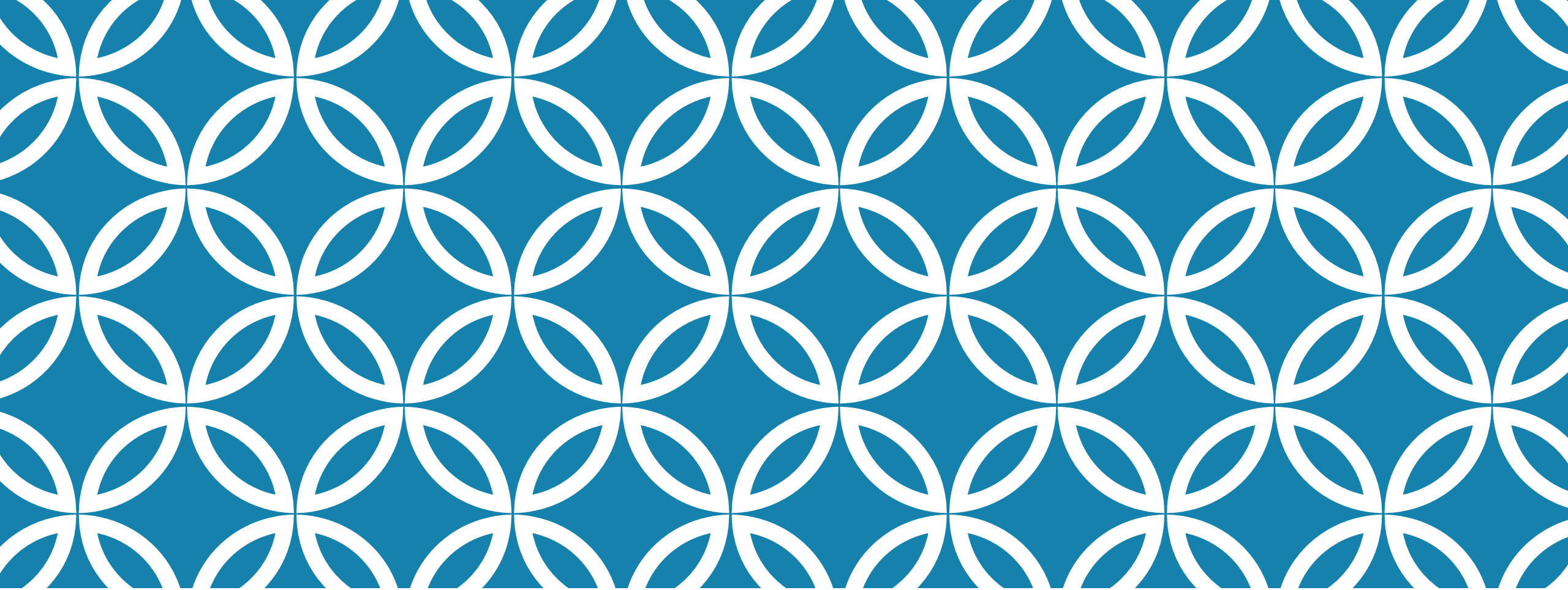
- Issue in Proximity Calculation
  - How to handle the case in which attributes have different scales and/or are correlated:
    - Standardization and Correlation for Distance Measures
  - How to calculate proximity between objects that are composed of different types of attributes, e.g., quantitative and qualitative
    - Combining Similarities for heterogeneous Attributes
  - How to handle proximity calculation when attributes have different weights
- Selecting Right Proximity Measure

MEASURE OF SIMILARITY AND DISSIMILARITY:

- Selecting Right Proximity Measure
  - Proximity measure should fit the type of data
  - Euclidean (dense/continuous/metric distance)
  - Similarity measures that ignore 0-0 matches (sparse data)
  - magnitude of the time series data (Euclidean n correlation which uses built-in normalization)
  - transformation and normalization for proper similarity
  - For efficiency triangular inequality

THANK YOU...

Here we end with Unit - 1



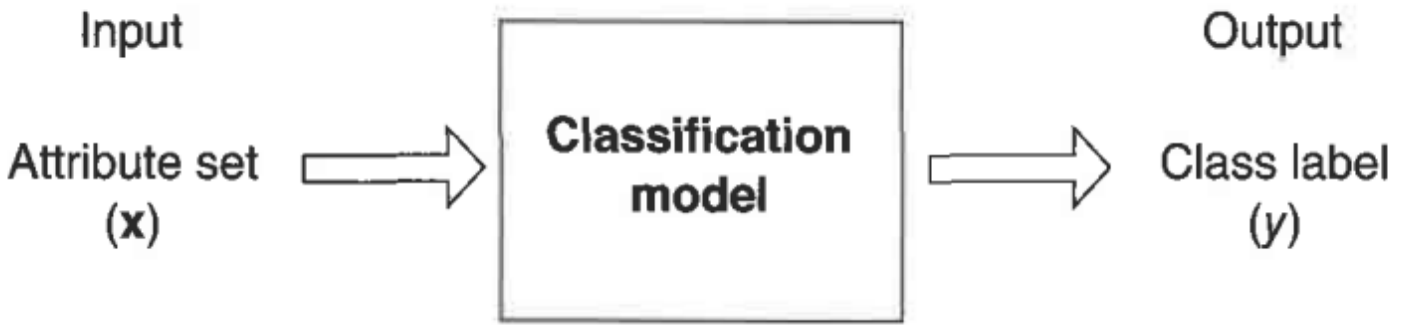
DATA MINING – UNIT 2 (SUPERVISED LEARNING)

By Rashmi Bhattad

UNIT-2

- Classification: Preliminaries;
- General approach to solving a classification problem;
- Decision tree induction;
- Rule-based classifier;
- Multilinear and Logistic Regression.

CLASSIFICATION



Classification is the task of learning a target function / that maps each attribute set x to one of the predefined class Labels y.

CLASSIFICATION

- Classification Preliminaries:
  - Descriptive Modeling
  - Predictive Modeling
- General approach to solve a classification problem
- Decision Tree, Rule-based, Neural Network, Support Vector Machines, Naïve Bayes.

GENERAL APPROACH TO SOLVE A CLASSIFICATION PROBLEM

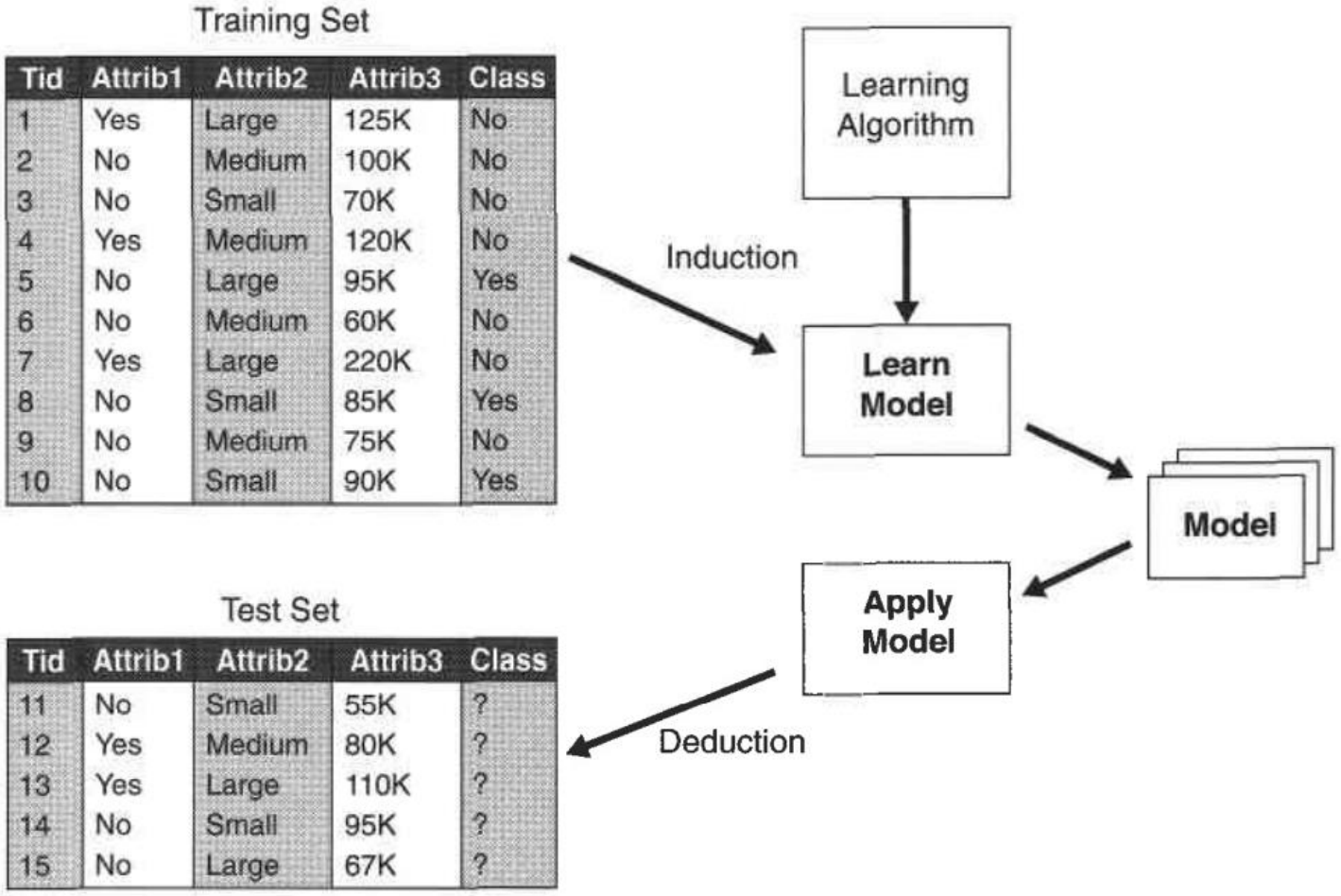
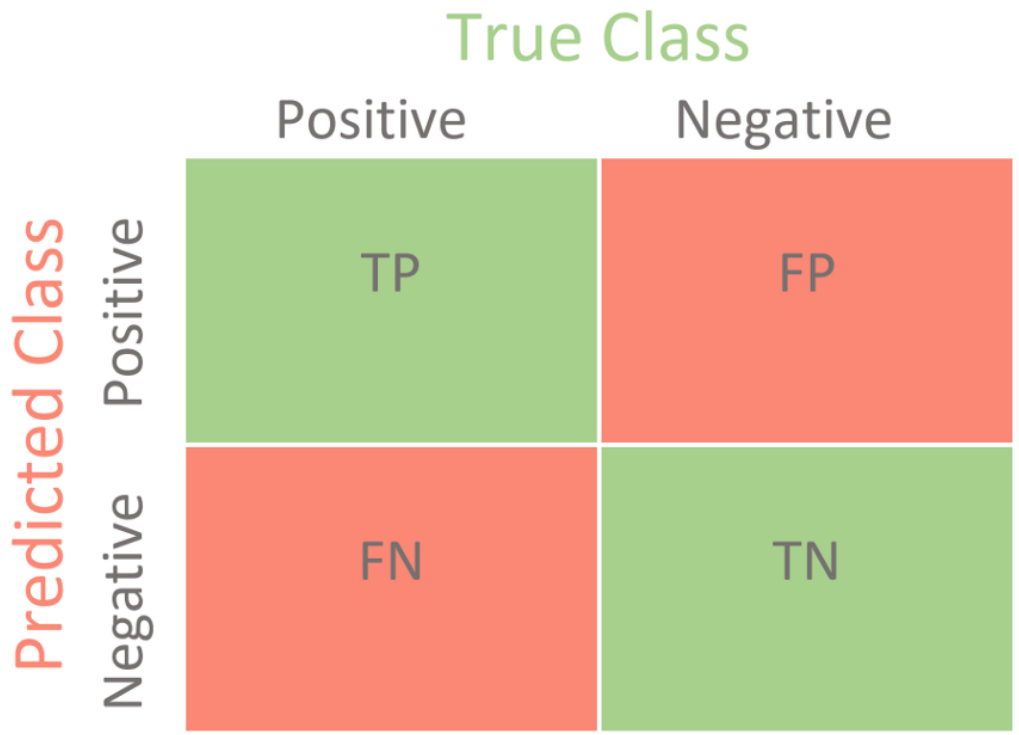


Figure 4.3. General approach for building a classification model.

CONFUSION MATRIX/CONTINGENCY TABLE:



$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall\ /\ Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

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

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

CONFUSION MATRIX/CONTINGENCY TABLE:

A data scientist is evaluating different binary classification models. A false positive result is 5 times more expensive (from a business perspective) than a false negative result.

The models should be evaluated based on the following criteria:

1) **Must have a recall rate of at least 80%**
2) **Must have a false positive rate of 10% or less**
3) **Must minimize business costs**

After creating each binary classification model, the data scientist generates the corresponding confusion matrix.

Which confusion matrix represents the model that satisfies the requirements?

A) TN = 91, FP = 9
FN = 22, TP = 78

B) TN = 99, FP = 1
FN = 21, TP = 79

C) TN = 96, FP = 4
FN = 10, TP = 90

D) TN = 98, FP = 2
FN = 18, TP = 82

CONFUSION MATRIX/CONTINGENCY TABLE:

Recall = TP / (TP + FN)

False Positive Rate (FPR) = FP / (FP + TN)

Cost = 5 * FP + FN

	A	B	C	D
Recall	78 / (78 + 22) = 0.78	79 / (79 + 21) = 0.79	90 / (90 + 10) = 0.9	82 / (82 + 18) = 0.82
False Positive Rate	9 / (9 + 91) = 0.09	1 / (1 + 99) = 0.01	4 / (4 + 96) = 0.04	2 / (2 + 98) = 0.02
Costs	5 * 9 + 22 = 67	5 * 1 + 21 = 26	5 * 4 + 10 = 30	5 * 2 + 18 = 28

ACCURACY FOR MULTI CLASS:

For following multi-class classification predictions:

	Predicted		
	15	2	3
Actual	7	15	8
	2	3	45

Calculate Accuracy, Per Class Precision, Per Class Recall.

Prove with an example FP=Neg-TN.

DECISION TREE:

- How the decision tree works:
  - Root Node, Internal Node, and Leaf Node
- How to build a decision tree
- Decision tree: ID3, C4.5, and CART (Hunt's algorithm)
- Design issues in decision tree induction
  - How should the training records be split?
  - How should the splitting procedure stop?
- Next

BUILDING DECISION TREE USING HUNT'S ALGO:

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training set for predicting borrowers who will default on loan payments.

BUILDING DECISION TREE USING HUNT'S ALGO:

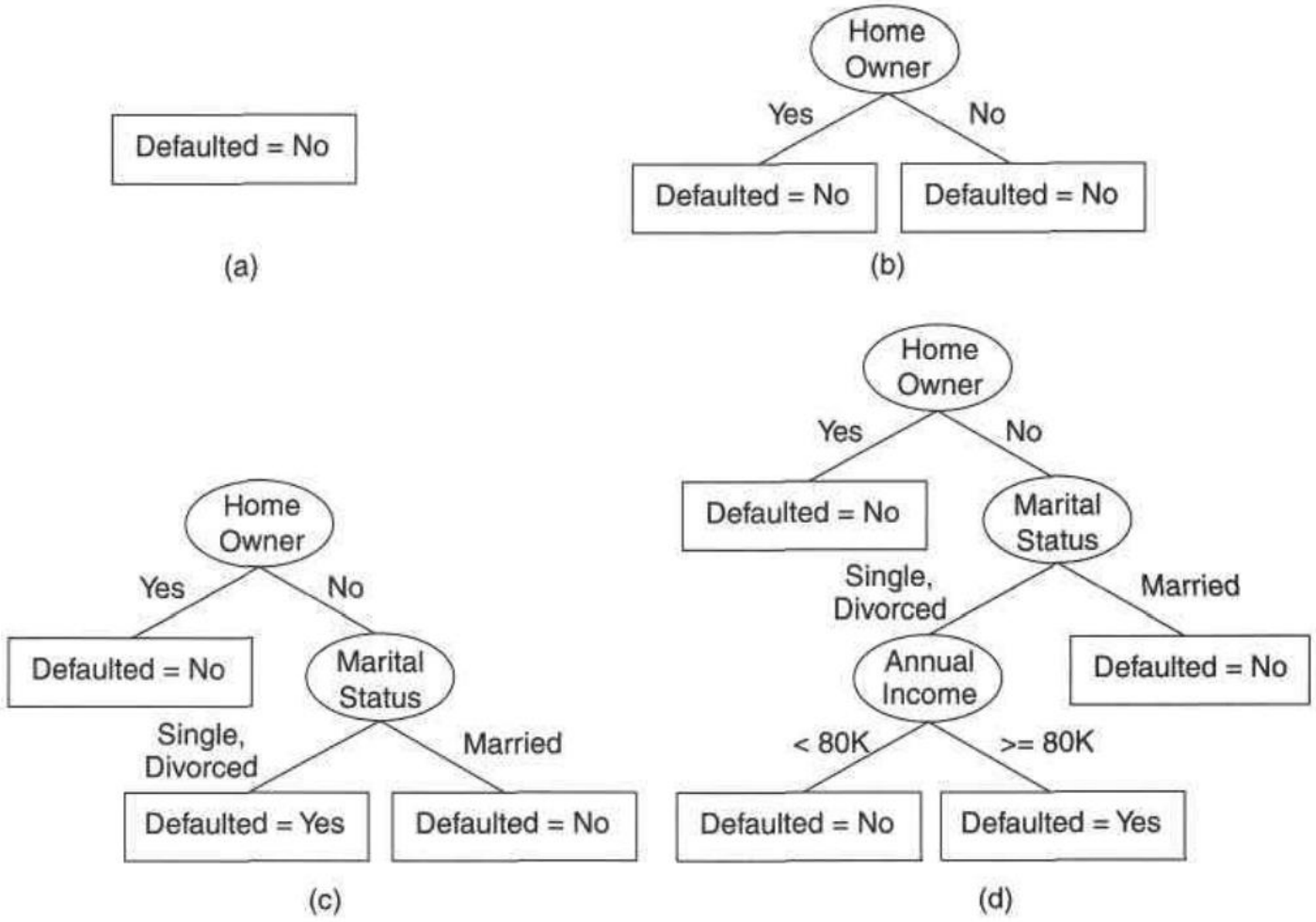


Figure 4.7. Hunt's algorithm for inducing decision trees.

BUILD DECISION TREE:

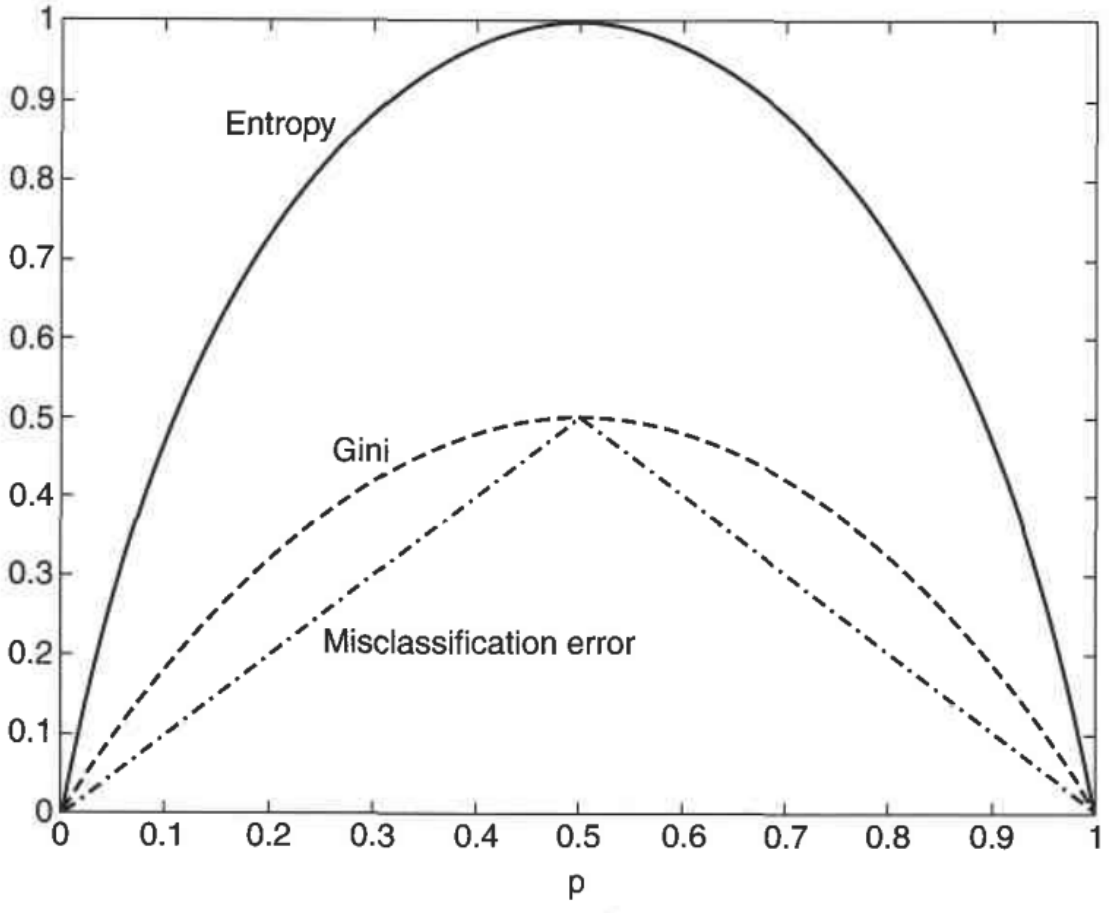
Day	Outlook	Temperature	Humidity	Wind	Play cricket
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

IMPURITY MEASURES INCLUDE:

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t),$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2,$$

$$\text{Classification error}(t) = 1 - \max_i [p(i|t)],$$



IMPURITY MEASURES INCLUDE:

Node N_1	Count
Class=0	0
Class=1	6

$$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$$
$$\text{Entropy} = -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$$
$$\text{Error} = 1 - \max[0/6, 6/6] = 0$$

Node N_2	Count
Class=0	1
Class=1	5

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$
$$\text{Entropy} = -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.650$$
$$\text{Error} = 1 - \max[1/6, 5/6] = 0.167$$

Node N_3	Count
Class=0	3
Class=1	3

$$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$$
$$\text{Entropy} = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$$
$$\text{Error} = 1 - \max[3/6, 3/6] = 0.5$$

IMPURITY MEASURES INCLUDE:

- Information Gain:
  - Compare degree of impurity of parent node (before splitting) with impurity of child node (after splitting)

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j),$$

- Gain Ratio:

$$\text{Gain ratio} = \frac{\Delta_{\text{info}}}{\text{Split Info}}.$$

$$\text{Split Info} = -\sum_{i=1}^{k} P(v_i) \log_2 P(v_i)$$

ID3

$$Values\,(Outlook) = Sunny, Overcast, Rain$$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$

$S_{Sunny} \leftarrow [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5} log_2 \frac{2}{5} - \frac{3}{5} log_2 \frac{3}{5} = 0.971$

$S_{Overcast} \leftarrow [4+, 0-]$ $\quad Entropy(S_{Overcast}) = -\frac{4}{4} log_2 \frac{4}{4} - \frac{0}{4} log_2 \frac{0}{4} = 0$

$S_{Rain} \leftarrow [3+, 2-]$ $\quad Entropy(S_{Rain}) = -\frac{3}{5} log_2 \frac{3}{5} - \frac{2}{5} log_2 \frac{2}{5} = 0.971$

$$Gain\,(S, Outlook) = Entropy(S) - \sum_{v \in \{Sunny, Overcast, Rain\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Outlook)$$

$$= Entropy(S) - \frac{5}{14} Entropy(S_{Sunny}) - \frac{4}{14} Entropy(S_{Overcast})$$

$$- \frac{5}{14} Entropy(S_{Rain})$$

$$Gain(S, Outlook) = 0.94 - \frac{5}{14} 0.971 - \frac{4}{14}\ 0 - \frac{5}{14} 0.971 = 0.2464$$

Attribute: Temp

$$Values\,(Temp) = Hot, Mild, Cool$$

$$S = [9+, 5-] \qquad Entropy(S) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

$$S_{Hot} \leftarrow [2+, 2-] \qquad Entropy(S_{Hot}) = -\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4} = 1.0$$

$$S_{Mild} \leftarrow [4+, 2-] \qquad Entropy(S_{Mild}) = -\frac{4}{6}log_2\frac{4}{6} - \frac{2}{6}log_2\frac{2}{6} = 0.9183$$

$$S_{Cool} \leftarrow [3+, 1-] \qquad Entropy(S_{Cool}) = -\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4} = 0.8113$$

$$Gain\,(S, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Temp)$$

$$= Entropy(S) - \frac{4}{14}Entropy(S_{Hot}) - \frac{6}{14}Entropy(S_{Mild})$$

$$- \frac{4}{14}Entropy(S_{Cool})$$

Attribute: Humidity

$$Values\,(Humidity) = High, Normal$$

$$S = [9+, 5-] \qquad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$$

$$S_{High} \leftarrow [3+, 4-] \qquad Entropy(S_{High}) = -\frac{3}{7} log_2 \frac{3}{7} - \frac{4}{7} log_2 \frac{4}{7} = 0.9852$$

$$S_{Normal} \leftarrow [6+, 1-] \qquad Entropy(S_{Normal}) = -\frac{6}{7} log_2 \frac{6}{7} - \frac{1}{7} log_2 \frac{1}{7} = 0.5916$$

$$Gain\,(S, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Humidity)$$

$$= Entropy(S) - \frac{7}{14} Entropy(S_{High}) - \frac{7}{14} Entropy(S_{Normal})$$

$$Gain(S, Humidity) = 0.94 - \frac{7}{14} 0.9852 - \frac{7}{14}\, 0.5916 = 0.\ 1516$$

Attribute: Wind

$$Values\,(Wind) = Strong, Weak$$

$$S = [9+, 5-] \qquad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$$

$$S_{Strong} \leftarrow [3+, 3-] \qquad Entropy(S_{Strong}) = 1.0$$

$$S_{Weak} \leftarrow [6+, 2-] \qquad Entropy(S_{Weak}) = -\frac{6}{8} log_2 \frac{6}{8} - \frac{2}{8} log_2 \frac{2}{8} = 0.8113$$

$$Gain\,(S, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Wind) = Entropy(S) - \frac{6}{14} Entropy(S_{Strong}) - \frac{8}{14} Entropy(S_{Weak})$$

$$Gain(S, Wind) = 0.94 - \frac{6}{14}\ 1.0 - \frac{8}{14}\ 0.8113 = 0.0478$$

COMPARING INFORMATION GAIN:

$$Gain(S, Outlook) = 0.2464$$

$$Gain(S, Temp) = 0.0289$$

$$Gain(S, Humidity) = 0.1516$$

$$Gain(S, Wind) = 0.0478$$

SUNNY:

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Attribute: Temp

$$Values\ (Temp) = Hot, Mild, Cool$$

$$S_{Sunny} = [2+, 3-] \qquad Entropy(S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$$

$$S_{Hot} \leftarrow [0+, 2-] \qquad Entropy(S_{Hot}) = 0.0$$

$$S_{Mild} \leftarrow [1+, 1-] \qquad Entropy(S_{Mild}) = 1.0$$

$$S_{Cool} \leftarrow [1+, 0-] \qquad Entropy(S_{Cool}) = 0.0$$

$$Gain\ (S_{Sunny}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Temp)$$

$$= Entropy(S) - \frac{2}{5}Entropy(S_{Hot}) - \frac{2}{5}Entropy(S_{Mild})$$

$$- \frac{1}{5}Entropy(S_{Cool})$$

$$Gain(S_{sunny}, Temp) = 0.97 - \frac{2}{5}0.0 - \frac{2}{5}\,1 - \frac{1}{5}0.0 = 0.570$$

SUNNY:

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Attribute: Humidity

$$Values\,(Humidity) = High, Normal$$

$$S_{Sunny} = [2+, 3-] \qquad Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$$

$$S_{high} \leftarrow [0+, 3-] \qquad Entropy(S_{High}) = 0.0$$

$$S_{Normal} \leftarrow [2+, 0-] \qquad Entropy(S_{Normal}) = 0.0$$

$$Gain\,(S_{Sunny}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Humidity) = Entropy(S) - \frac{3}{5}Entropy(S_{High}) - \frac{2}{5}Entropy(S_{Normal})$$

$$Gain(S_{sunny}, Humidity) = 0.97 - \frac{3}{5}\,0.0 - \frac{2}{5}0.0 = 0.97$$

SUNNY:

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Attribute: Wind

$$Values\,(Wind) = Strong, Weak$$

$$S_{Sunny} = [2+, 3-] \qquad Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$$

$$S_{Strong} \leftarrow [1+, 1-] \qquad Entropy(S_{Strong}) = 1.0$$

$$S_{Weak} \leftarrow [1+, 2-] \qquad Entropy(S_{Weak}) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = 0.9183$$

$$Gain\,(S_{Sunny}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

$$Gain(S_{sunny}, Wind) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}\,0.918 = 0.0192$$

SUNNY:

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

$$Gain(S_{sunny}, Temp) = 0.570$$

$$Gain(S_{sunny}, Humidity) = 0.97$$

$$Gain(S_{sunny}, Wind) = 0.0192$$

RAIN:

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Temp

$$Values\,(Temp) = Hot, Mild, Cool$$

$$S_{Rain} = [3+, 2-] \qquad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$$

$$S_{Hot} \leftarrow [0+, 0-] \qquad Entropy(S_{Hot}) = 0.0$$

$$S_{Mild} \leftarrow [2+, 1-] \qquad Entropy(S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$$

$$S_{Cool} \leftarrow [1+, 1-] \qquad Entropy(S_{Cool}) = 1.0$$

$$Gain\,(S_{Rain}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Temp)$$

$$= Entropy(S) - \frac{0}{5}Entropy(S_{Hot}) - \frac{3}{5}Entropy(S_{Mild})$$

$$- \frac{2}{5}Entropy(S_{Cool})$$

$$Gain(S_{Rain}, Temp) = 0.97 - \frac{0}{5}0.0 - \frac{3}{5}\,0.918 - \frac{2}{5}\,1.0 = 0.0192$$

RAIN:

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Humidity

$$Values\,(Humidity) = High, Normal$$

$$S_{Rain} = [3+, 2-] \qquad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$$

$$S_{High} \leftarrow [1+, 1-] \qquad Entropy(S_{High}) = 1.0$$

$$S_{Normal} \leftarrow [2+, 1-] \qquad Entropy(S_{Normal}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$$

$$Gain\,(S_{Rain}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Humidity) = Entropy(S) - \frac{2}{5}Entropy(S_{High}) - \frac{3}{5}Entropy(S_{Normal})$$

$$Gain(S_{Rain}, Humidity) = 0.97 - \frac{2}{5}\,1.0 - \frac{3}{5}0.918 = 0.0192$$

RAIN:

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Wind

$$Values\ (wind) = Strong, Weak$$

$$S_{Rain} = [3+, 2-] \qquad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$$

$$S_{Strong} \leftarrow [0+, 2-] \qquad Entropy(S_{Strong}) = 0.0$$

$$S_{Weak} \leftarrow [3+, 0-] \qquad Entropy(S_{weak}) = 0.0$$

$$Gain\ (S_{Rain}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

$$Gain(S_{Rain}, Wind) = 0.97 - \frac{2}{5}\ 0.0 - \frac{3}{5}0.0 = 0.97$$

$$Gain(S_{Rain}, Temp) = 0.0192$$

$$Gain(S_{Rain}, Humidity) = 0.0192$$

GINI:

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

The gain in gini after splitting on A is:

$$G_{A=T} = 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = 0.4898$$

$$G_{A=F} = 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

$$\Delta = G_{orig} - 7/10 G_{A=T} - 3/10 G_{A=F} = 0.1371$$

The gain in gini after splitting on B is:

$$G_{B=T} = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = 0.3750$$

$$G_{B=F} = 1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2 = 0.2778$$

$$\Delta = G_{orig} - 4/10 G_{B=T} - 6/10 G_{B=F} = 0.1633$$

Therefore, attribute B will be chosen to split the node.

DECISION TREE INDUCTION ALGORITHM:

Algorithm 4.1 A skeleton decision tree induction algorithm.

```
TreeGrowth (E, F)
 1: if stopping_cond(E,F) = true then
 2:    leaf = createNode().
 3:    leaf.label = Classify(E).
 4:    return leaf.
 5: else
 6:    root = createNode().
 7:    root.test_cond = find_best_split(E, F).
 8:    let V = {v|v is a possible outcome of root.test_cond }.
 9:    for each v ∈ V do
10:       E_v = {e | root.test_cond(e) = v and e ∈ E}.
11:       child = TreeGrowth(E_v, F).
12:       add child as descendent of root and label the edge (root → child) as v.
13:    end for
14: end if
15: return root.
```

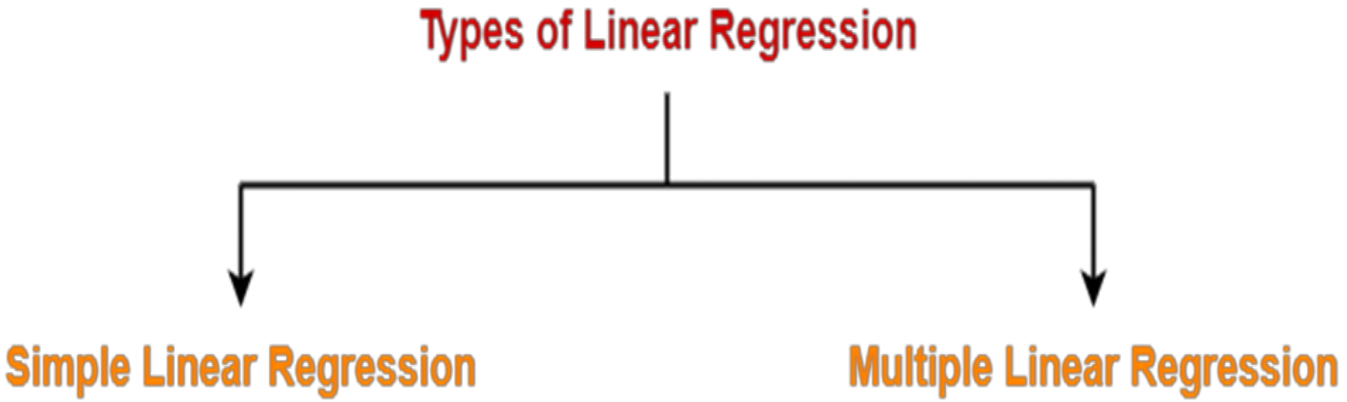
LINEAR REGRESSION ANALYSIS:

In Machine Learning,

- Linear Regression is a supervised machine learning algorithm.
- It tries to find out the best linear relationship that describes the data you have.
- It assumes that there exists a linear relationship between a dependent variable and independent variable(s).
- The value of the dependent variable of a linear regression model is a continuous value i.e. real numbers.

LINEAR REGRESSION ANALYSIS:

- Dependent variable: the variable we wish to explain
- Independent variable: the variable used to explain the dependent variable



SIMPLE LINEAR REGRESSION:

Only one independent variable (thus, simple), X

- Relationship between X and Y is described by a linear function
- Changes in Y are assumed to be caused by changes in X, that is,
  - Change In X Causes Change in Y

SIMPLE LINEAR REGRESSION:

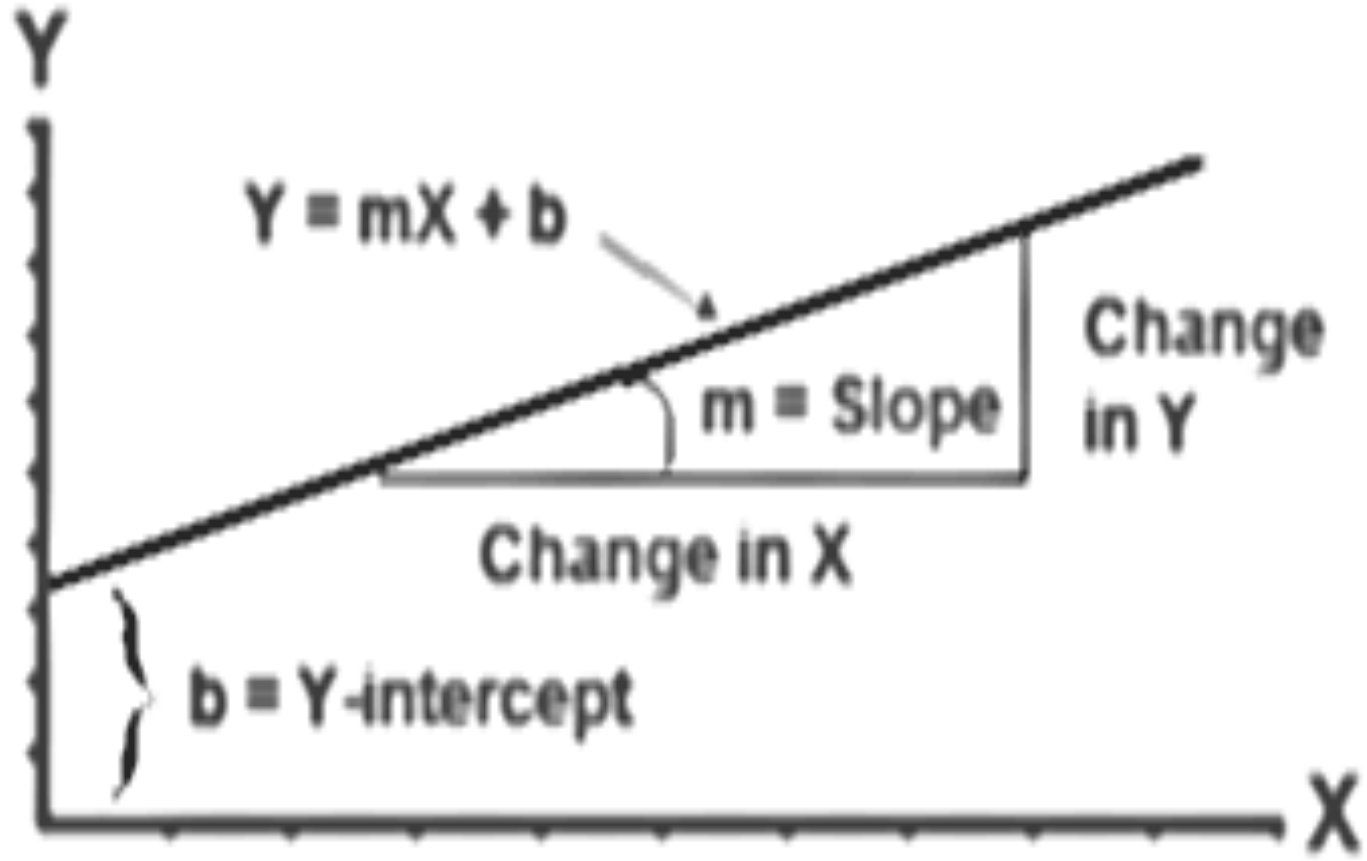
For simple linear regression, the form of the model is-

$$\mathbf{Y = \beta 0 + \beta 1 X}$$

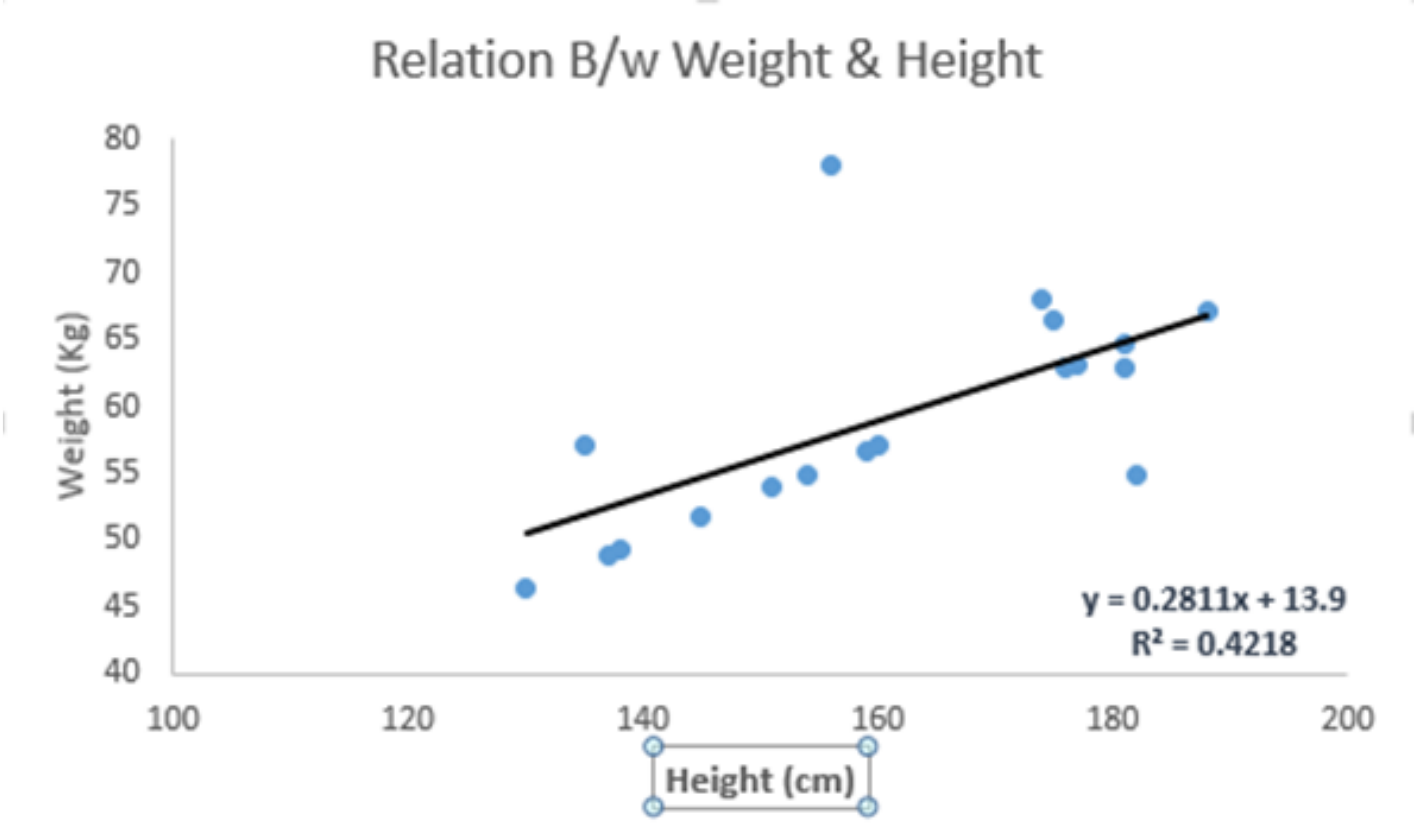
Here,

- Y is a dependent variable.
- X is an independent variable.
- β0 and β1 are the regression coefficients.
- β0 is the intercept or the bias that fixes the offset to a line.
- β1 is the slope or weight that specifies the factor by which X has an impact on Y.

SIMPLE LINEAR REGRESSION:



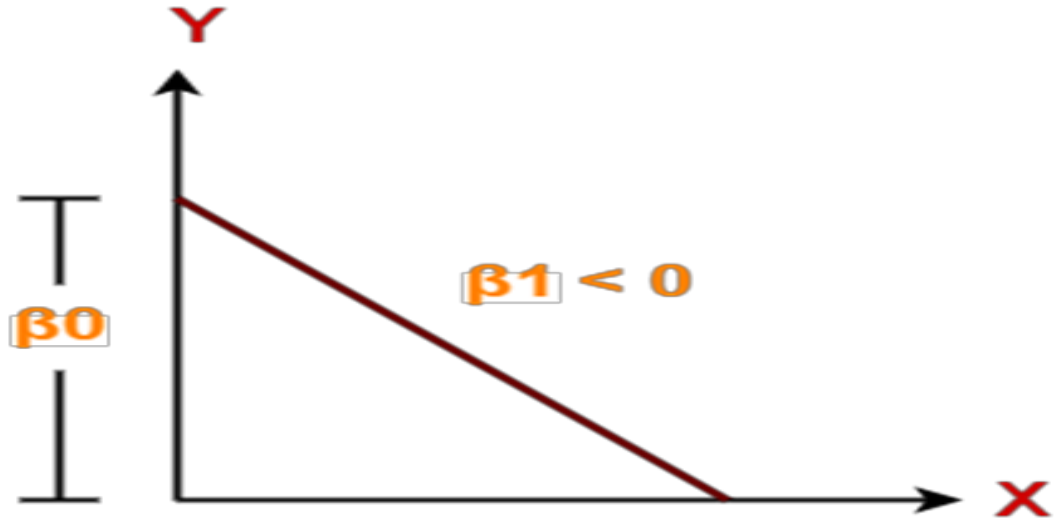
SIMPLE LINEAR REGRESSION:



SIMPLE LINEAR REGRESSION:

Case - 01: $\beta_1 < 0$

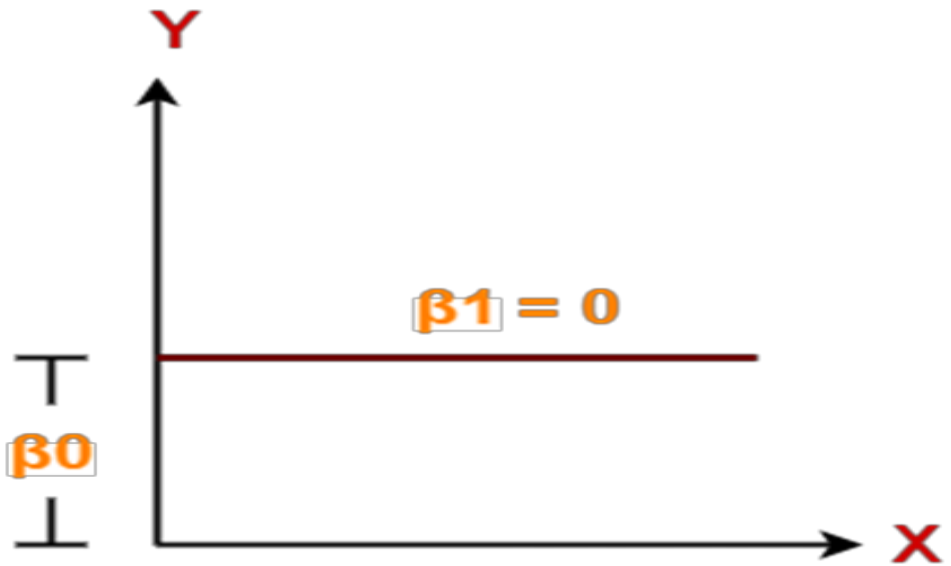
- It indicates that variable X has negative impact on Y.
- If X increases, Y will decrease and vice-versa.



SIMPLE LINEAR REGRESSION:

Case - 02: $\beta_1 = 0$

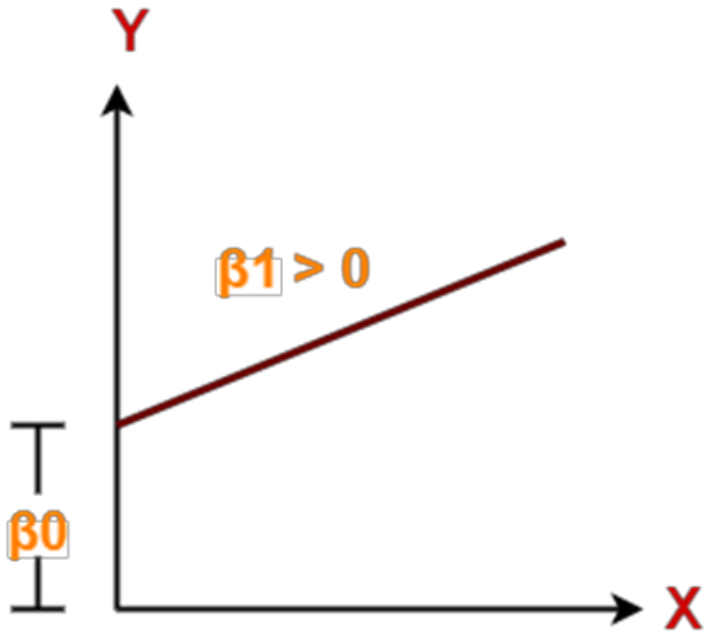
- It indicates that variable X has no impact on Y.
- If X changes, there will be no change in Y.



SIMPLE LINEAR REGRESSION:

Case - 03: $\beta_1 > 0$

- It indicates that variable X has positive impact on Y.
- If X increases, Y will increase and vice-versa.



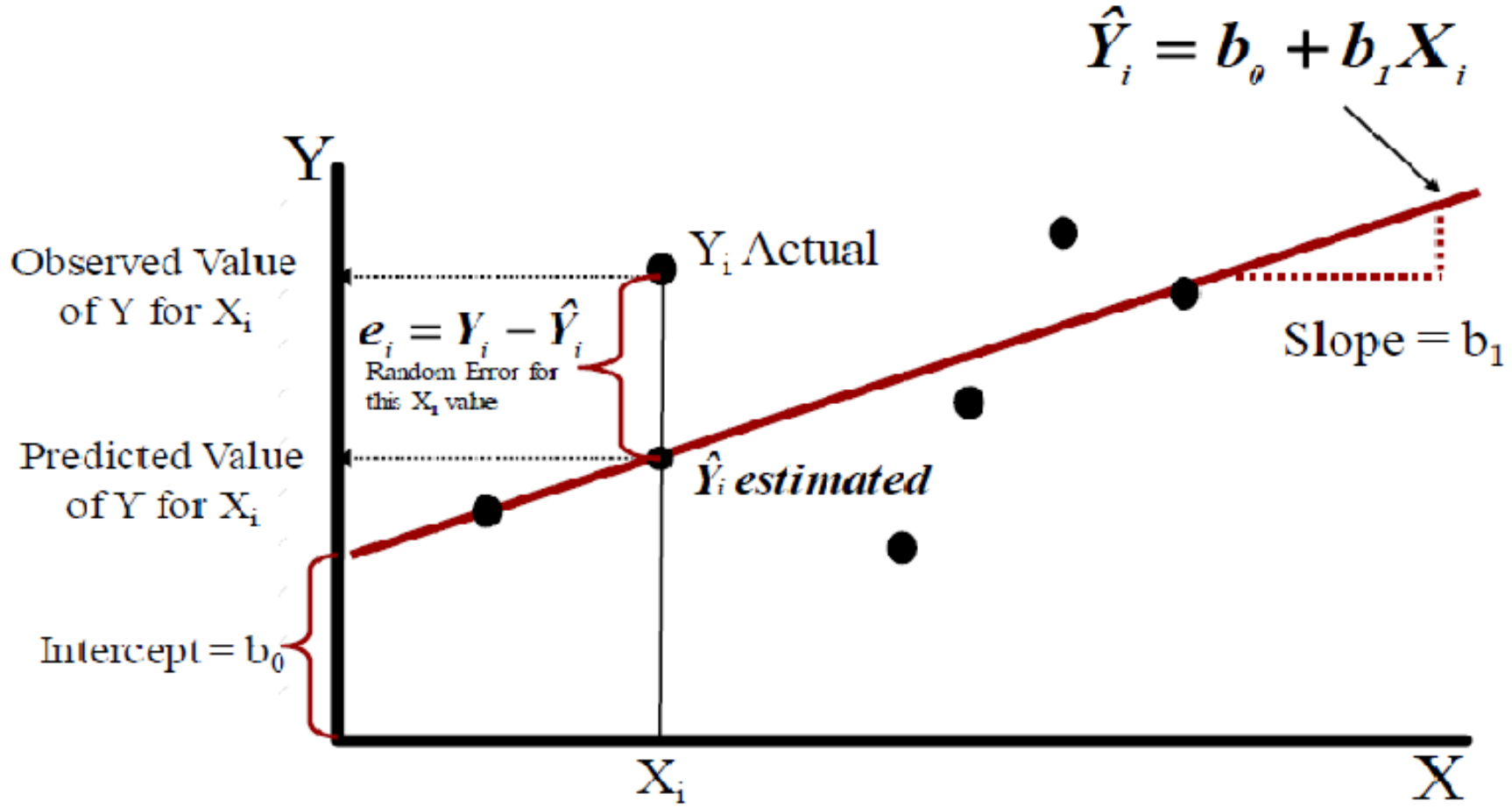
SIMPLE LINEAR REGRESSION:

- Ordinary Least Squared Method (OLS)
- We try to estimate the regression line such that the sum of squared differences are minimized, thus, the name Ordinary Least Squared Method (OLS)
- i.e.

$$min \sum e^2_i = min \sum (Y_i - \hat{Y}_i)^2 =$$

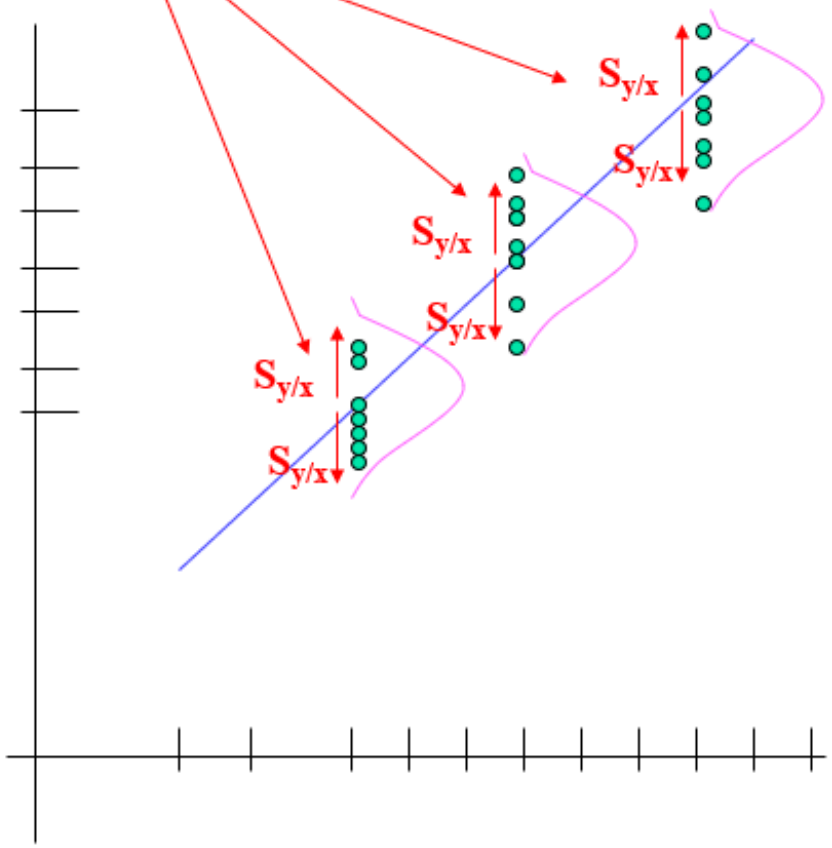
$$min \sum (Y_i - (b_0 + b_1 X_i))^2$$

SIMPLE LINEAR REGRESSION:

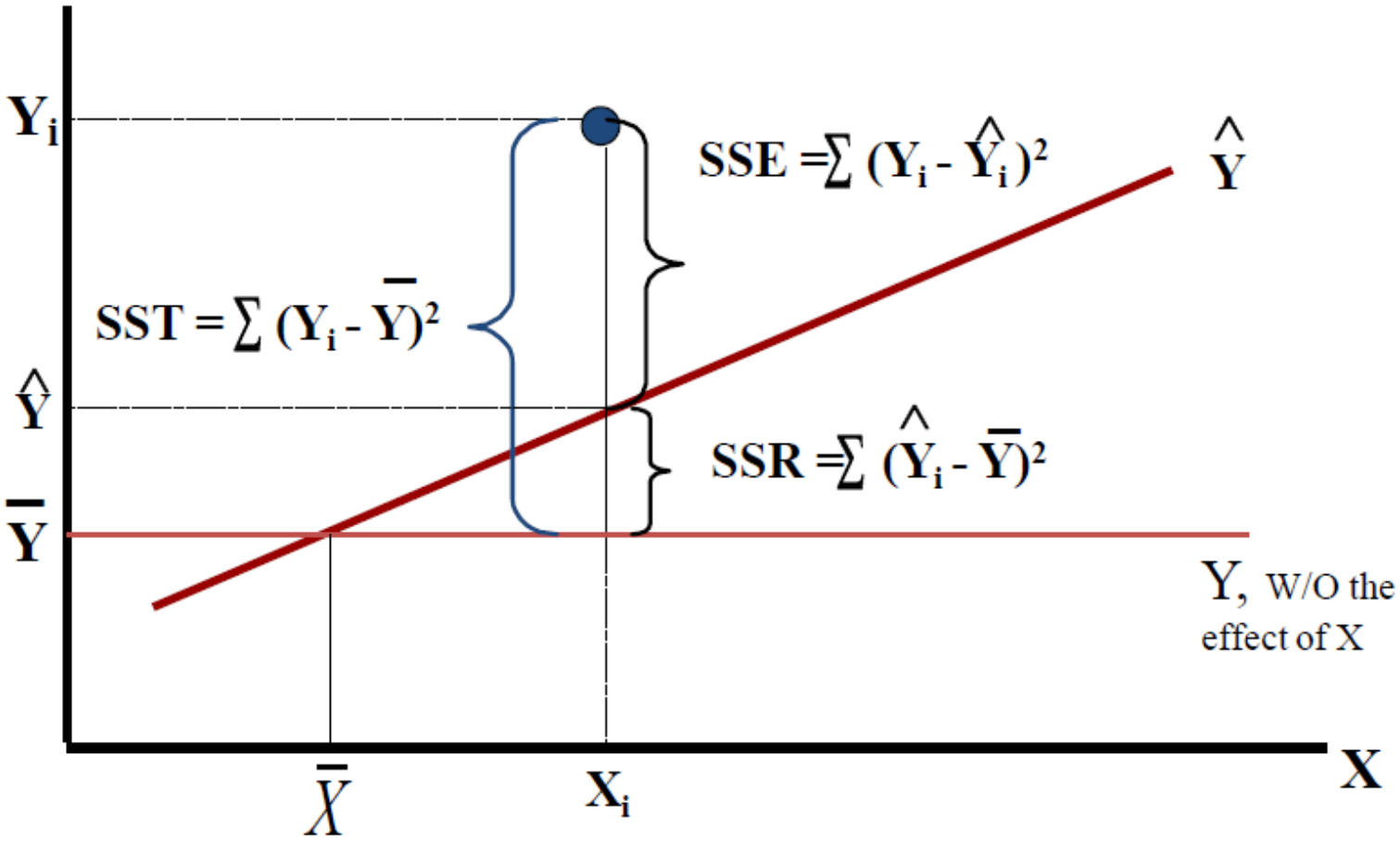


ASSESSING PERFORMANCE OF REGRESSION

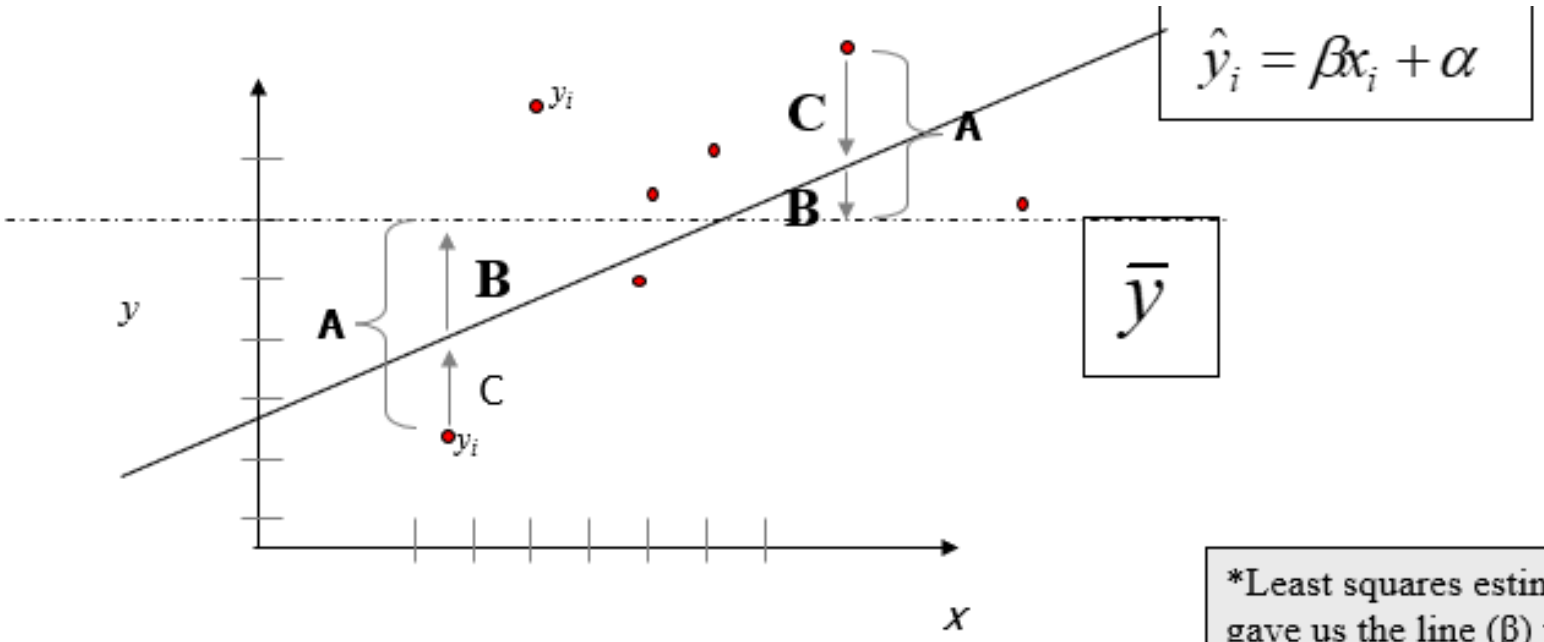
The standard error of Y given X is the average variability around the regression line at any given value of X. It is assumed to be equal at all values of X.



ASSESSING PERFORMANCE OF REGRESSION



ASSESSING PERFORMANCE OF REGRESSION



*Least squares estimation gave us the line (β) that minimized C^2

$$\sum_{i=1}^{n} (y_i - \bar{y})^2 = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

$R^2 = SS_{reg}/SS_{total}$

A^2

SS_{total}

Total squared distance of observations from naïve mean of y

Total variation

B^2

SS_{reg}

Distance from regression line to naïve mean of y

Variability due to x (regression)

C^2

$SS_{residual}$

Variance around the regression line

Additional variability not explained by x—what least squares method aims to minimize

ASSESSING PERFORMANCE OF REGRESSION – ERROR MEASURES

- Total variation is made up of two parts:

$$SST = SSR + SSE$$

Total Sum of Squares	Regression Sum of Squares	Error Sum of Squares

$$SST = \sum (Y_i - \bar{Y})^2 \quad SSR = \sum (\hat{Y}_i - \bar{Y})^2 \quad SSE = \sum (Y_i - \hat{Y}_i)^2$$

where:

$\bar{Y}$ = Average value of the dependent variable

Y_i = Observed values of the dependent variable

$\hat{Y}_i$ = Predicted value of Y for the given X_i value

ASSESSING PERFORMANCE OF REGRESSION:

- The coefficient of determination is the portion of the total variation in the dependent variable, Y, that is explained by variation in the independent variable, X
- The coefficient of determination is also called r-squared and is denoted as r^2.

$$r^2 = \frac{SSR}{SST} = \frac{regression\ sum\ of\ squares}{total\ sum\ of\ squares}$$

$$0 \leq r^2 \leq 1$$

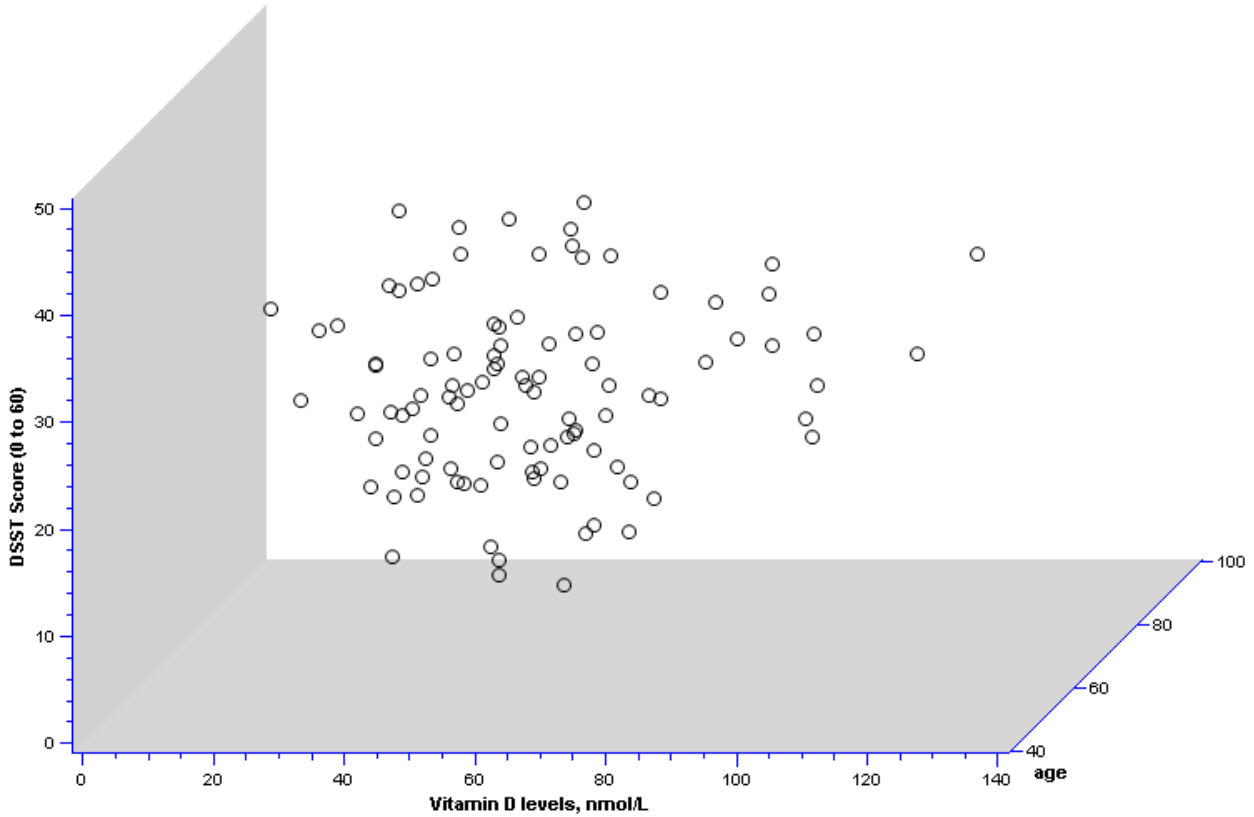
RESULTING FORMULAS:

Slope (beta coefficient) = $\hat{\beta} = \frac{Cov(x, y)}{Var(x)}$

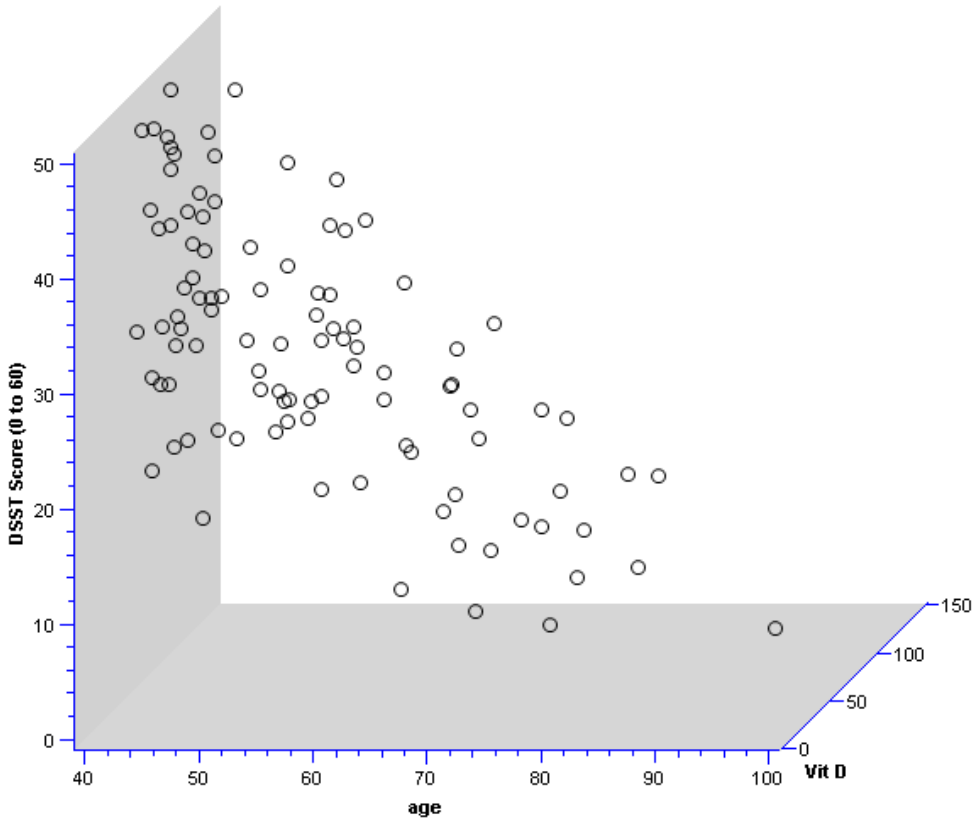
Intercept= Calculate $: \hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$

Regression line always goes through the point: $(\bar{x}, \bar{y})$

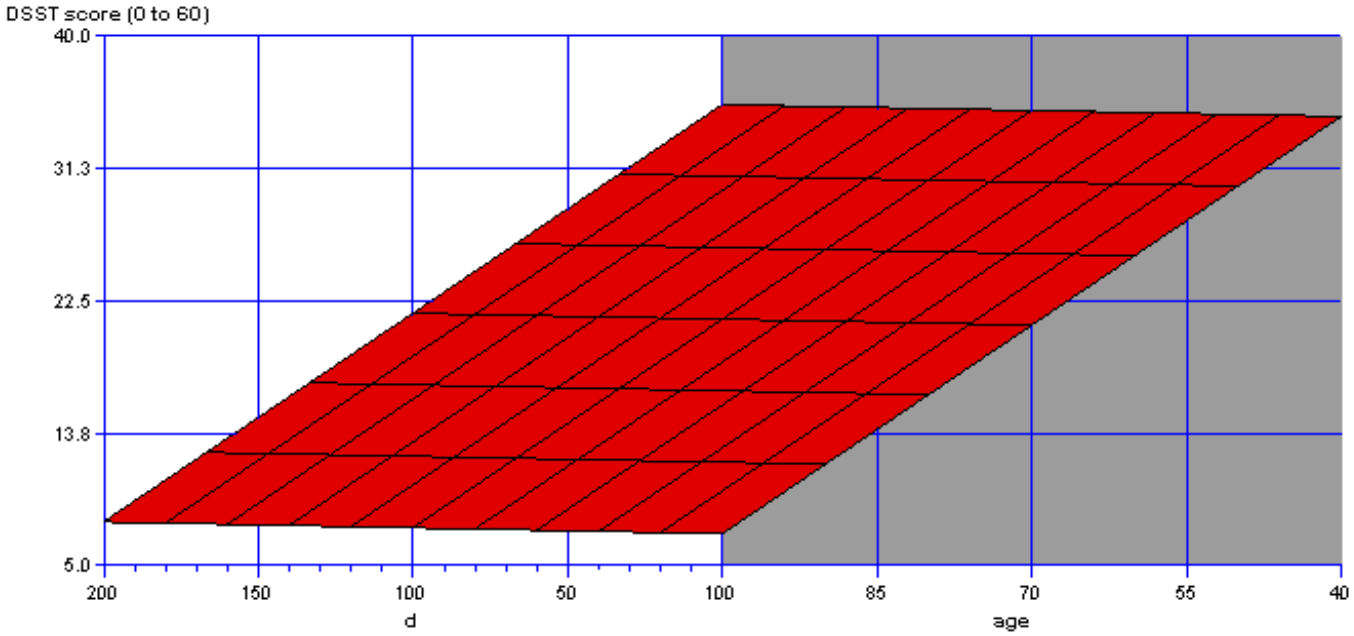
MULTIVARIATE REGRESSION:



MULTIVARIATE REGRESSION:



MULTIVARIATE REGRESSION:



On the plane, the slope for vitamin D is the same at every age; thus, the slope for vitamin D represents the effect of vitamin D when age is held constant.

MULTIVARIATE REGRESSION:

More than one predictor…

$E(y) = \alpha + \beta_1 * X + \beta_2 * W + \beta_3 * Z \ldots$

Each regression coefficient is the amount of change in the outcome variable that would be expected per one-unit change of the predictor, if all other variables in the model were held constant.

ASSESSING PERFORMANCE OF REGRESSION – ERROR MEASURES

- Sum of Squared Error (SSE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Goodness-fit (R2)
- Bias
- Variance

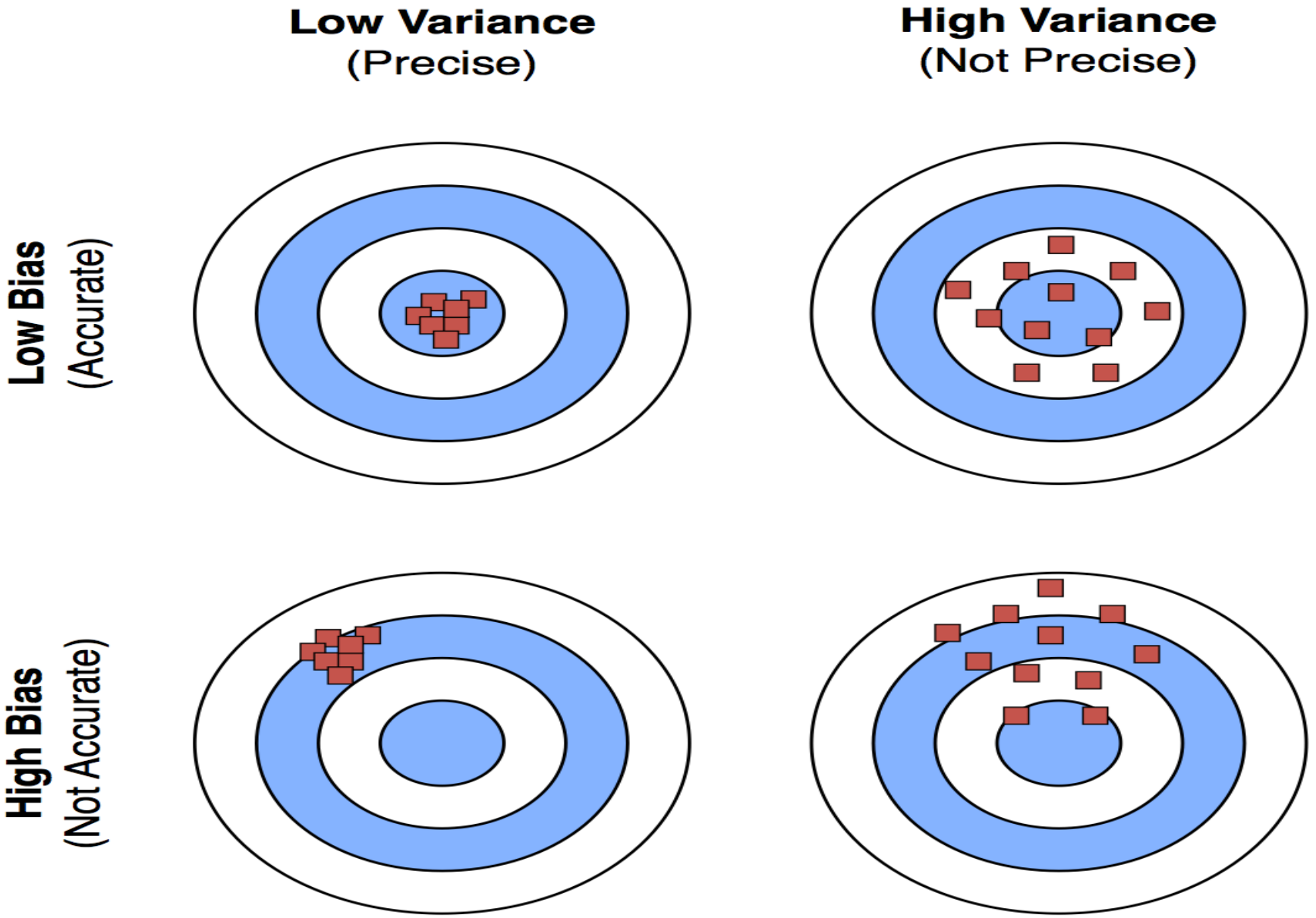
BIAS:

- The error due to bias is taken as the difference between the expected (or average) prediction of the model and the correct value that we are trying to predict.
- **Low Bias:** It means fewer assumptions about the form of the target function.
- **High-Bias:** It means more assumptions about the form of the target function.
- Examples of low-bias machine learning algorithms are **Decision Trees, Support Vector Machines, and k-Nearest Neighbors.**
- Examples of high-bias machine learning algorithms include **Linear Regression, Logistic Regression.**

VARIANCE:

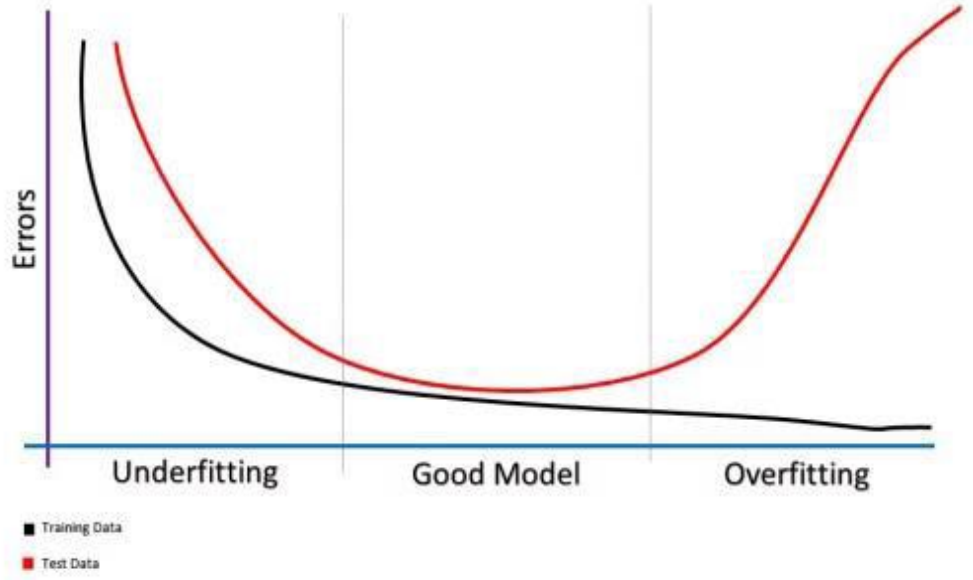
- Difference between what the model has learned from particular dataset and what the model was expected to learn
- **Low Variance**: It means small changes to the estimate of the target function with changes to the training dataset.
- **High Variance**: It means large changes to the estimate of the target function with changes to the training dataset.
- Examples of **low-variance machine learning algorithms** include: **Linear Regression, Logistic Regression.**
- Examples **of high-variance machine learning algorithms** include: **Decision Trees, Support Vector Machines and k-Nearest Neighbours**

BIAS VS VARIANCE:



OVERFITTING AND UNDERFITTING:

- Ein=In sample error:
- Error in Training Data
- Eout=Out of sample error or Generalization error or Validation error Error in Testing Data



OVERFITTING:

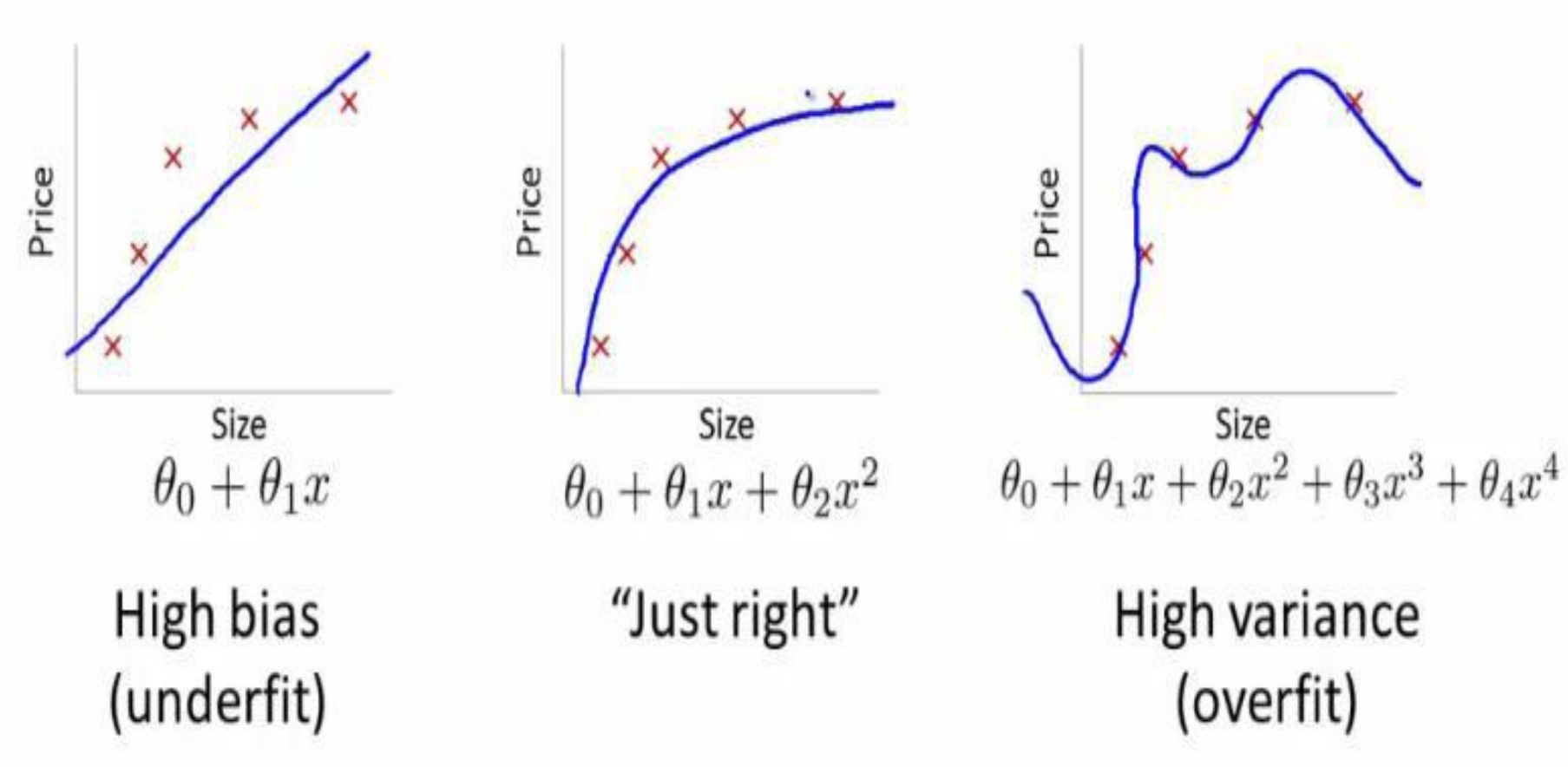
- **“Fitting the data more than necessary”**
- It makes - Model more complex
- It makes - Model uses additional degrees of freedom
- Validation error (Eout is high, training error Ein low)
- **low bias but high variance**

- Some methods to overcome overfitting,
  1. Cross Validation
  2. Reduce the model complexity
  3. Regularization

UNDERFITTING:

- “underfitting occurs when the model or the algorithm does not fit the data well enough”
- It makes - Model more simple
- Validation and training error high
- **low variance but high bias**

OVERFITTING AND UNDERFITTING:



COMMENT ON UF, OF, AND GF:

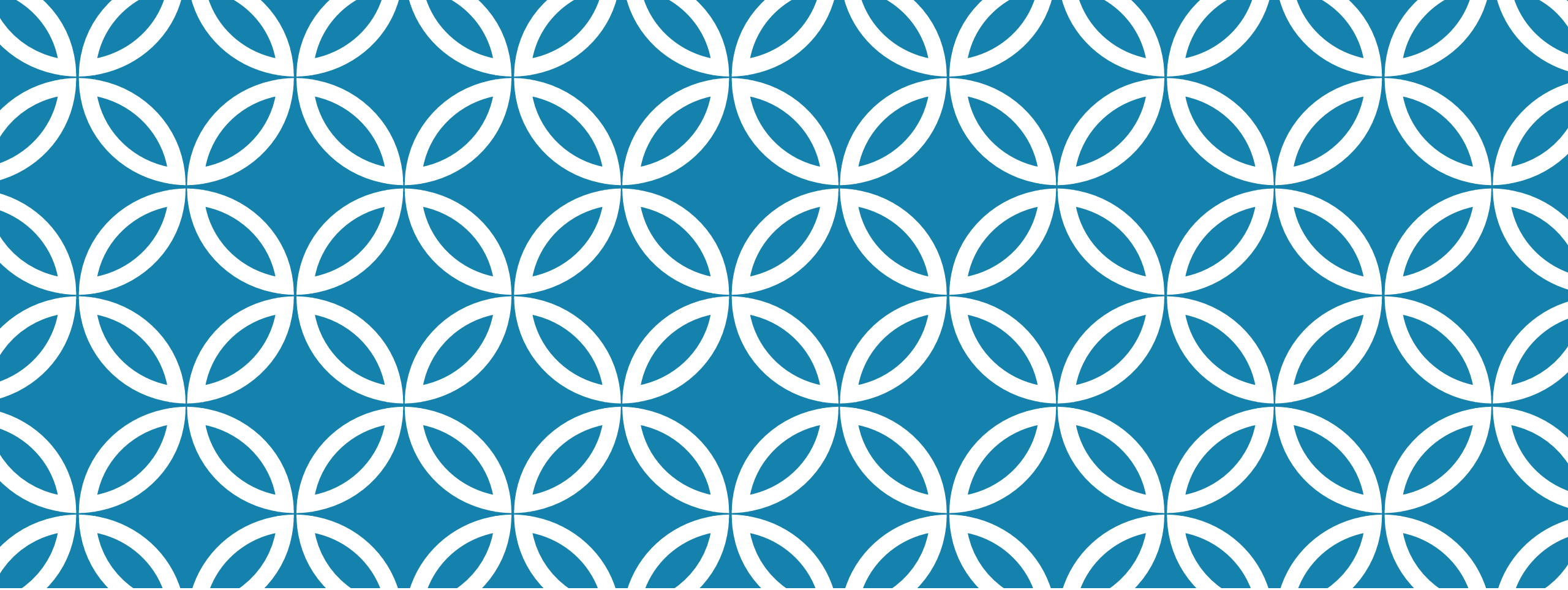
- Underfitting – Validation and training error high
- Overfitting – Validation error is high, training error low
- Good fit – Validation error low, slightly higher than the training error

COMMON MULTIVARIATE REGRESSION MODELS:

Outcome (dependent variable)	Example outcome variable	Appropriate multivariate regression model	Example equation	What do the coefficients give you?
Continuous	Blood pressure	**Linear regression**	blood pressure (mmHg) = α + βsalt*salt consumption (tsp/day) + βage*age (years) + βsmoker*ever smoker (yes=1/no=0)	slopes—tells you how much the outcome variable increases for every 1-unit increase in each predictor.
Binary	High blood pressure (yes/no)	**Logistic regression**	ln (odds of high blood pressure) = α + βsalt*salt consumption (tsp/day) + βage*age (years) + βsmoker*ever smoker (yes=1/no=0)	odds ratios—tells you how much the odds of the outcome increase for every 1-unit increase in each predictor.
Time-to-event	Time-to-death	**Cox regression**	ln (rate of death) = α + βsalt*salt consumption (tsp/day) + βage*age (years) + βsmoker*ever smoker (yes=1/no=0)	hazard ratios—tells you how much the rate of the outcome increases for every 1-unit increase in each predictor.

THANK YOU...

Here we end with Unit - 2



UNIT – 3
ASSOCIATION ANALYSIS

By Rashmi Bhattad

SYLLABUS:

Problem definition,

Frequent item set generation;

Rule Generation;

Compact representation of frequent item sets;

Alternative methods for generating frequent item sets.

FP-Growth algorithm,

Evaluation of association patterns,

Effect of skewed support distribution, Sequential patterns.

PROBLEM DEFINITION:

1. **Binary Representation of data:**

Table 6.2. A binary 0/1 representation of market basket data.

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

2. **Itemset and support count:**
   1. Itemset: Collection of zero or more items
   2. Support Count: Number of transactions that contain a particular itemset.

PROBLEM DEFINITION:

1. **Binary Representation of data:**
2. **Itemset and support count:**
   1. Itemset: Collection of zero or more items
   2. Support Count: Number of transactions that contain a particular itemset.
3. **Association Rule: x → y (x and y are disjoint set, they should have null intersection)**

   Support: how often rule is applicable to a given data set

   Confidence: how often items in Y appear in a transaction that contains item X.

$$\text{Support, } s(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{N};$$

$$\text{Confidence, } c(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}.$$

ASSOCIATION RULE DISCOVERY:

Definition Association Rule Discovery: Given a set of transactions T, find all the rules having support >= minsup and confidence >= minconf , where minsup and minconf are the corresponding support and confidence thresholds.

- Why Support and confidence are required?
  - Number of rules possible: $3^d-2^{d+1}+1$
  - Applying 20% minimum support and 50% confidence will remove more than 80% of data.
- Association rule mining divides the problem into two major subtasks:
  - Frequent Item Set Generation
  - Rule Generation

ASSOCIATION RULE MINING :

- Frequent Item Set Generation: whose objective is to find all the item sets that satisfy the minimum support threshold. These item sets are called frequent item sets.

- Rule Generation: whose objective is to extract all the high-confidence rules from the frequent item sets found in the previous step. These rules are called strong rules.

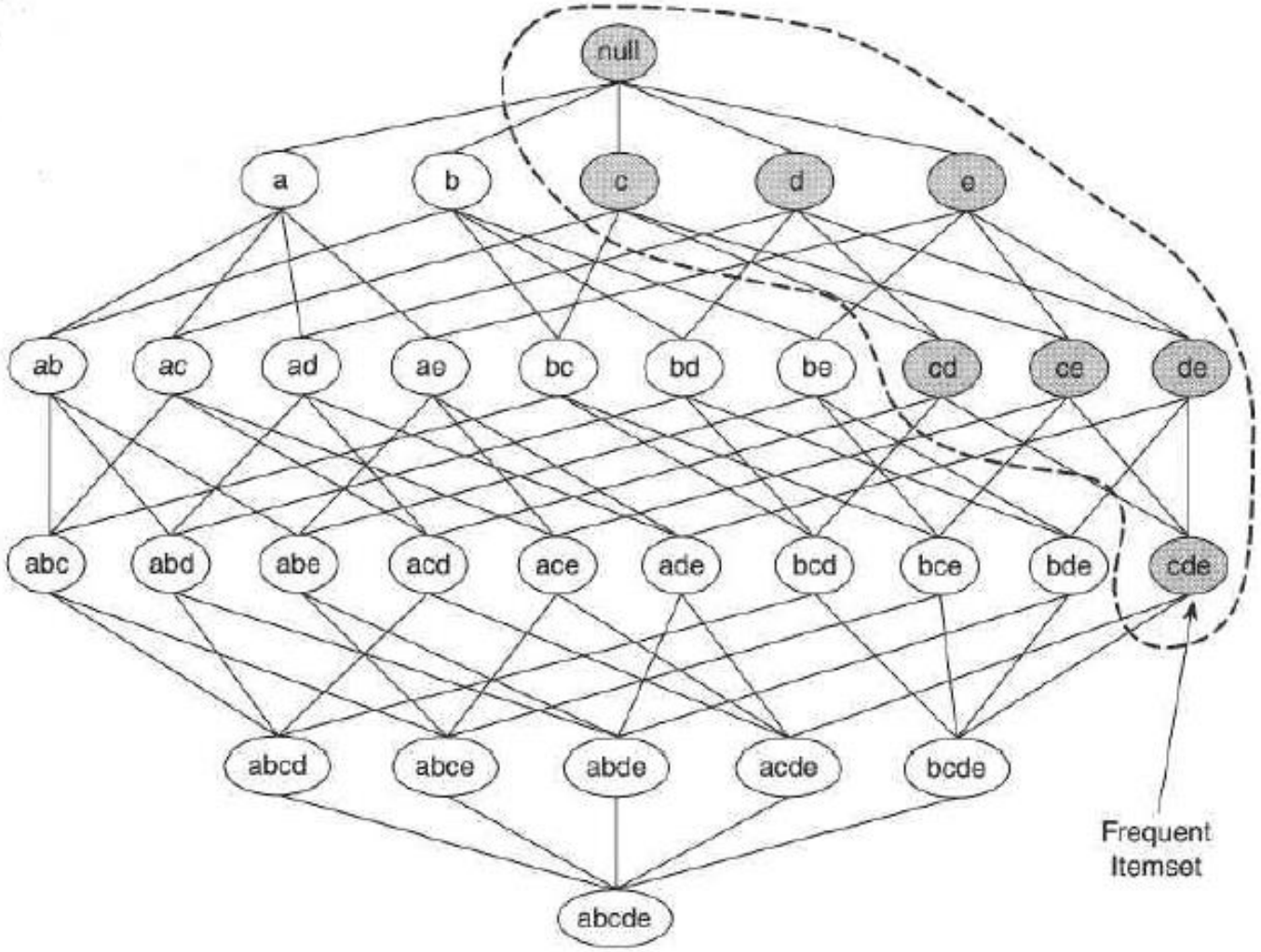
FREQUENT ITEM SET GENERATION:

- Frequent Item Set Generation: whose objective is to find all the item sets that satisfy the minimum support threshold.
- Dataset containing *k* items generates 2^k-1 frequent item sets.
- To reduce the computational complexity of frequent item set generation:
  - reduce the number of candidate item sets: **Apriori**
  - reduce the number of comparisons: Instead of matching each candidate itemset against every transaction, we can reduce the number of comparisons either by storing the candidate item sets or by compressing the data set.

APRIORI PRINCIPLE:

- If an itemset is frequent, then all of its subsets must also be frequent.
- Support based pruning
- Anti monotone property:

support(itemset) < support of its subsets



APRIORI:

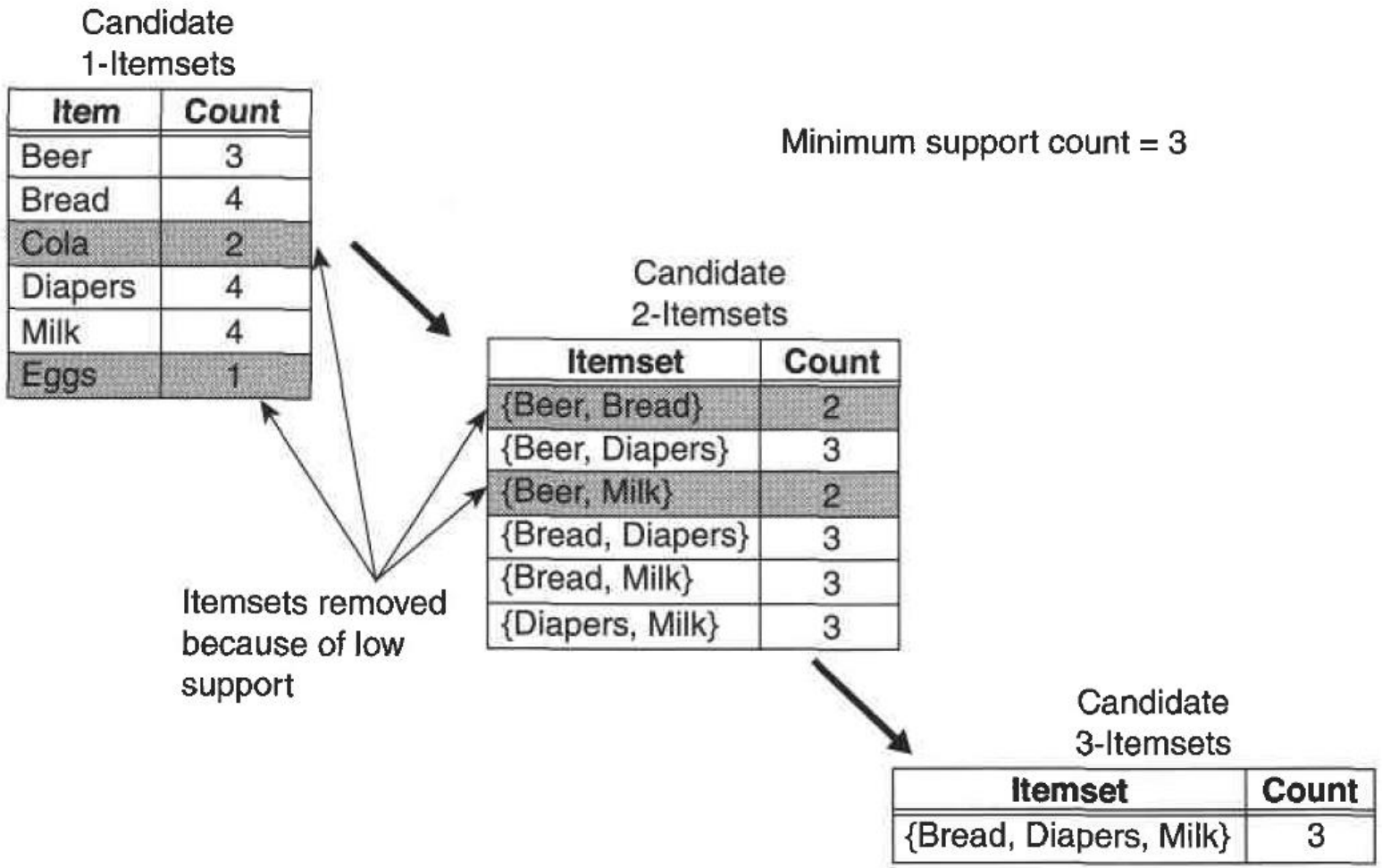


Figure 6.5. *Illustration of frequent itemset generation using the Apriori algorithm.*

APRIORI:

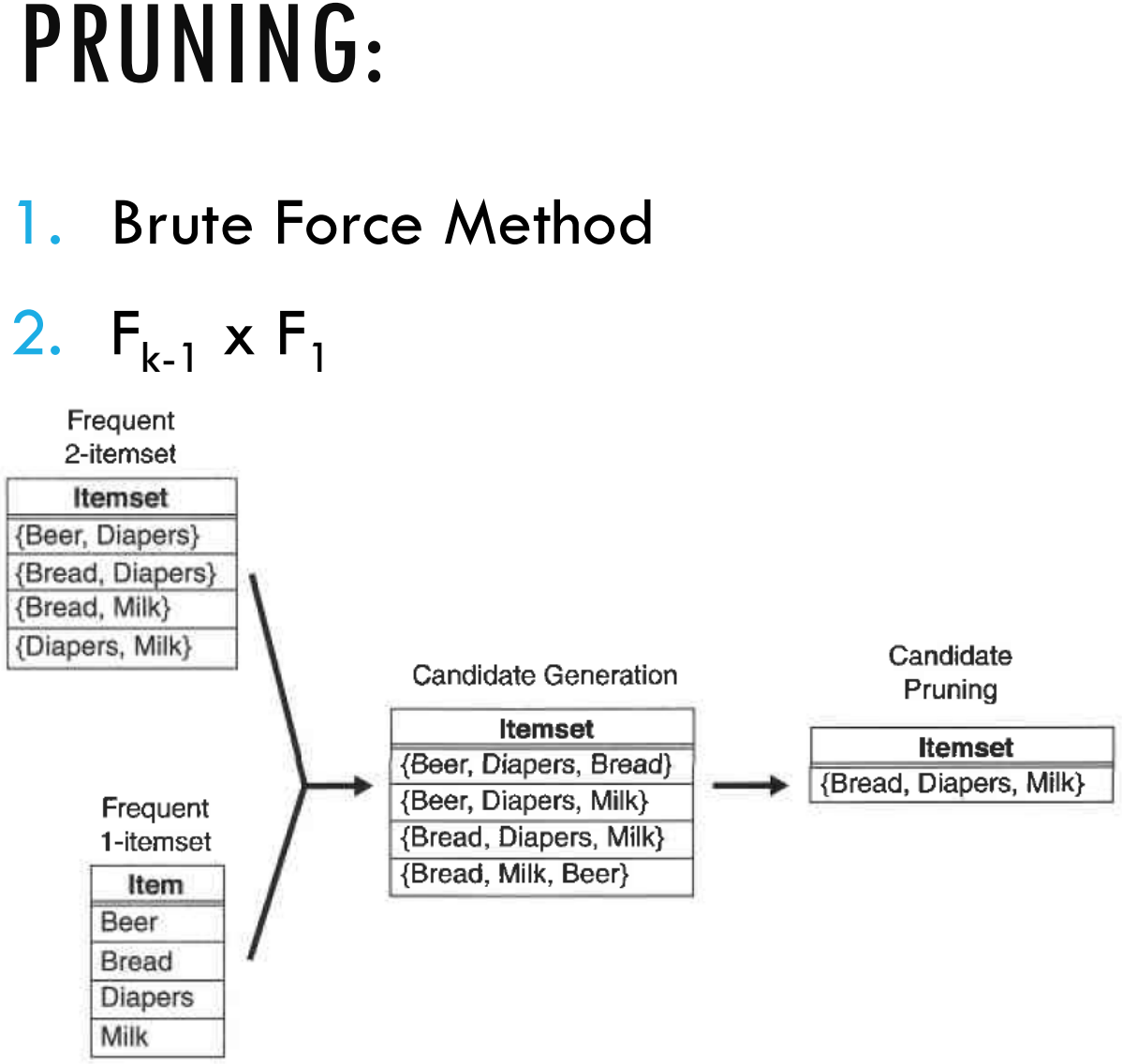
Algorithm 6.1 Frequent itemset generation of the *Apriori* algorithm.

1: $k = 1$.
2: $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times minsup \}$. {Find all frequent 1-itemsets}
3: **repeat**
4: $k = k + 1$.
5: $C_k = \text{apriori-gen}(F_{k-1})$. {Generate candidate itemsets}
6: **for** each transaction $t \in T$ **do**
7: $C_t = \text{subset}(C_k, t)$. {Identify all candidates that belong to t}
8: **for** each candidate itemset $c \in C_t$ **do**
9: $\sigma(c) = \sigma(c) + 1$. {Increment support count}
10: **end for**
11: **end for**
12: $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times minsup \}$. {Extract the frequent k-itemsets}
13: **until** $F_k = \emptyset$
14: Result $= \bigcup F_k$.

Candidate generation and pruning

APRIORI CANDIDATE KEY GENERATION AND PRUNING:

1. Brute Force Method
2. $F_{k-1} \times F_1$



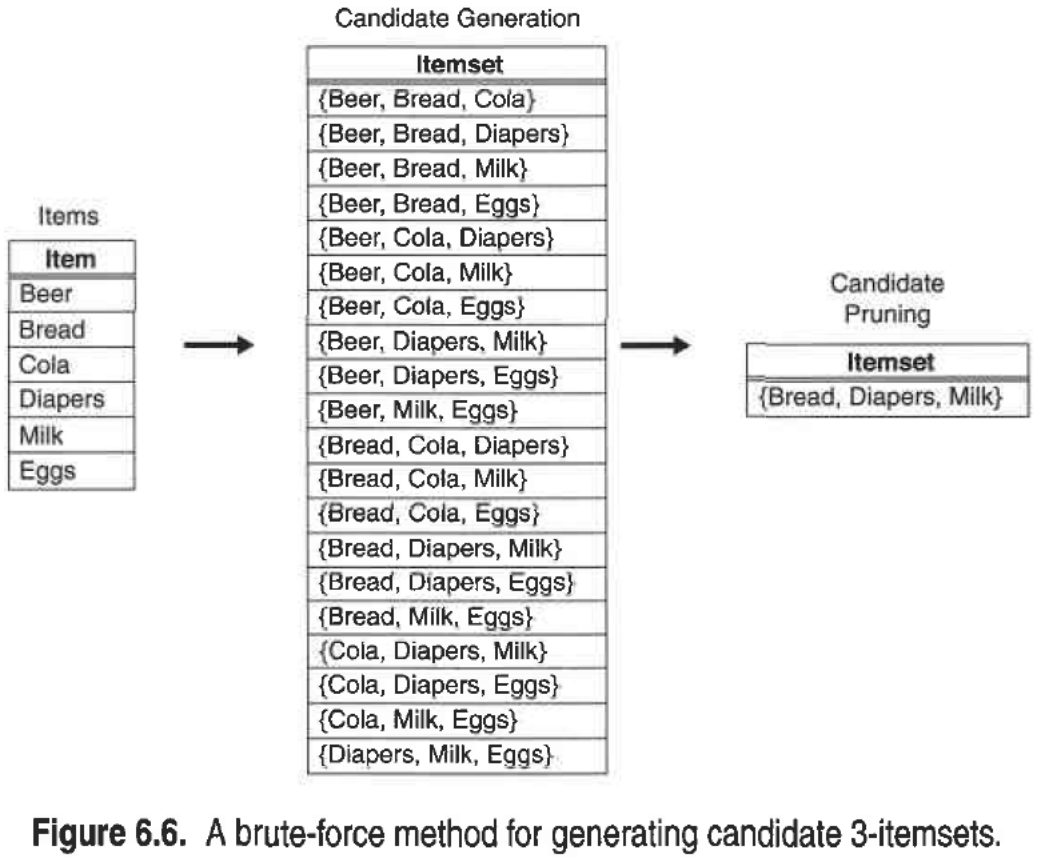


Figure 6.6. A brute-force method for generating candidate 3-itemsets.

APRIORI CANDIDATE KEY GENERATION AND PRUNING:

1. Brute Force Method
2. $F_{k-1} \times F_1$
3. $F_{k-1} \times F_{k-1}$

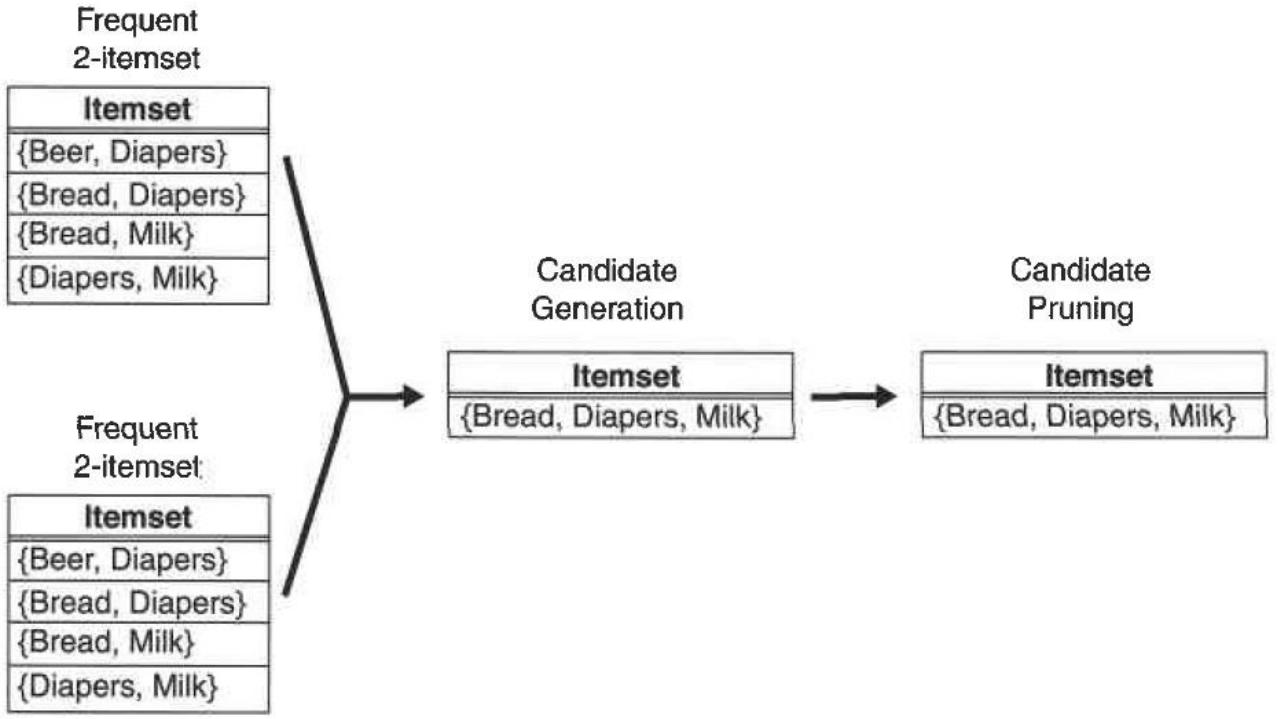


Figure 6.8. Generating and pruning candidate k-itemsets by merging pairs of frequent $(k-1)$-itemsets.

SUPPORT COUNTING:

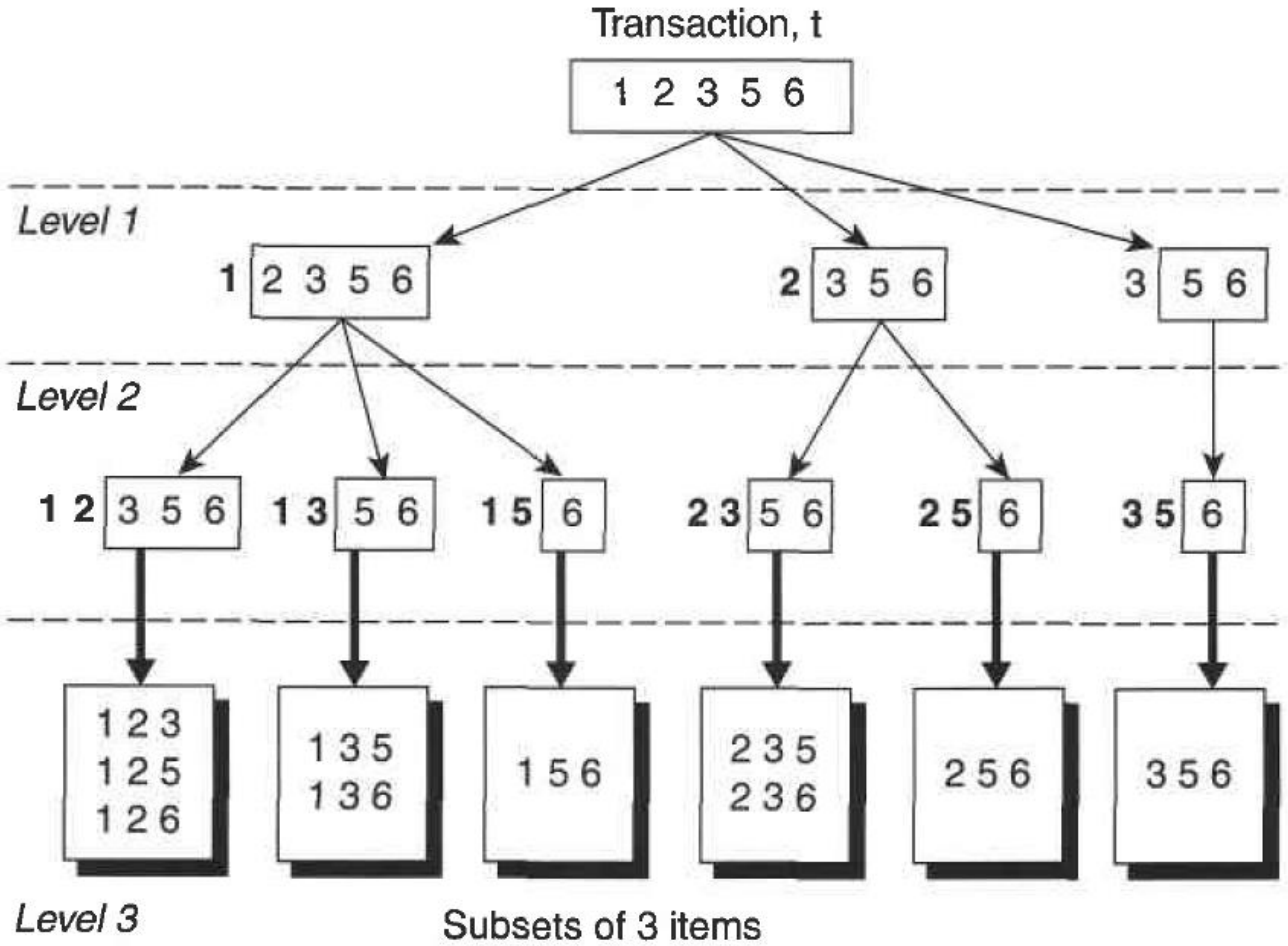


Figure 6.9. Enumerating subsets of three items from a transaction t.

APRIORI:

1. It is an array based algorithm.
2. It uses Join and Prune technique.
3. Apriori uses a breadth-first search
4. Apriori utilizes a **level-wise approach** where it generates patterns containing 1 item, then 2 items, then 3 items, and so on.
5. Candidate generation is extremely slow. Runtime increases exponentially depending on the number of different items.
6. Candidate generation is very parallelizable.
7. It requires large memory space due to large number of candidate generation.
8. It scans the database multiple times for **generating candidate sets**.

COMPUTATIONAL COMPLEXITY OF APRIORI:

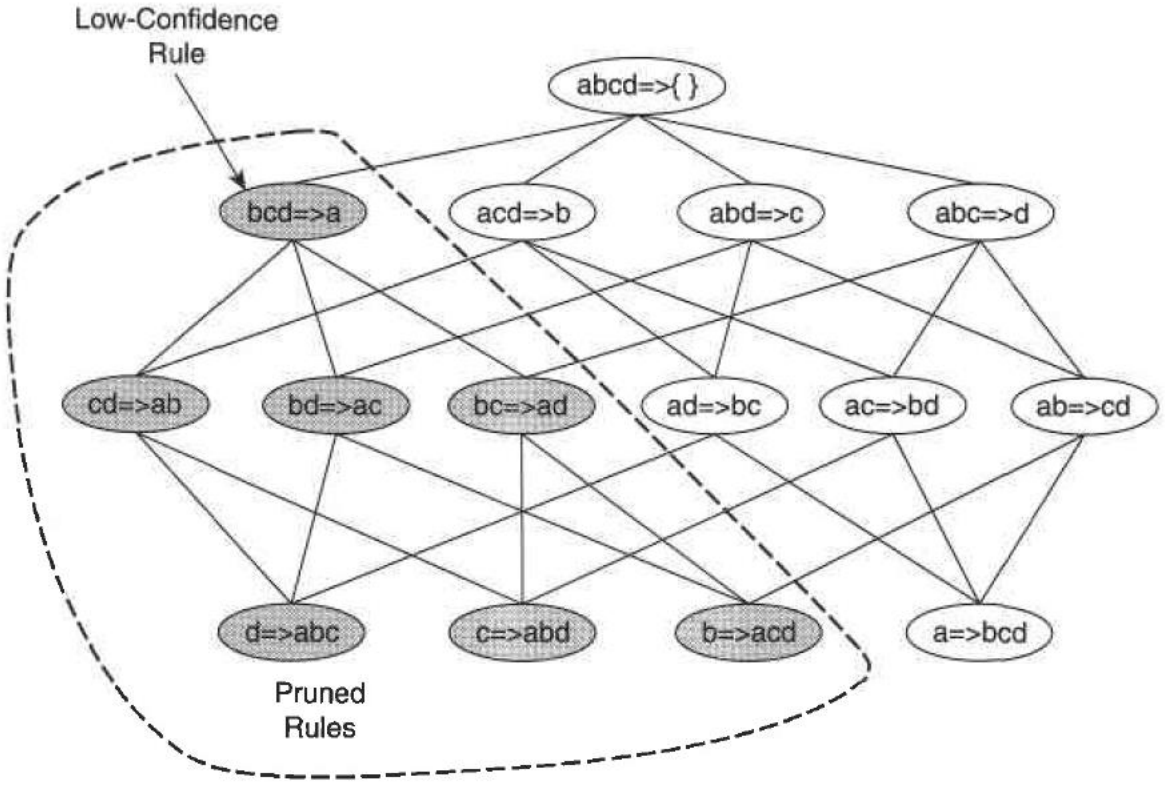
The computational complexity of Apriori gets affected by the following parameters:

1. Support Threshold
2. Number of Items (Dimensionality)
3. Number of transactions
4. Average transaction width
5. Generation of 1-frequent item set
6. Candidate generation
7. Support counting

RULE GENERATION:

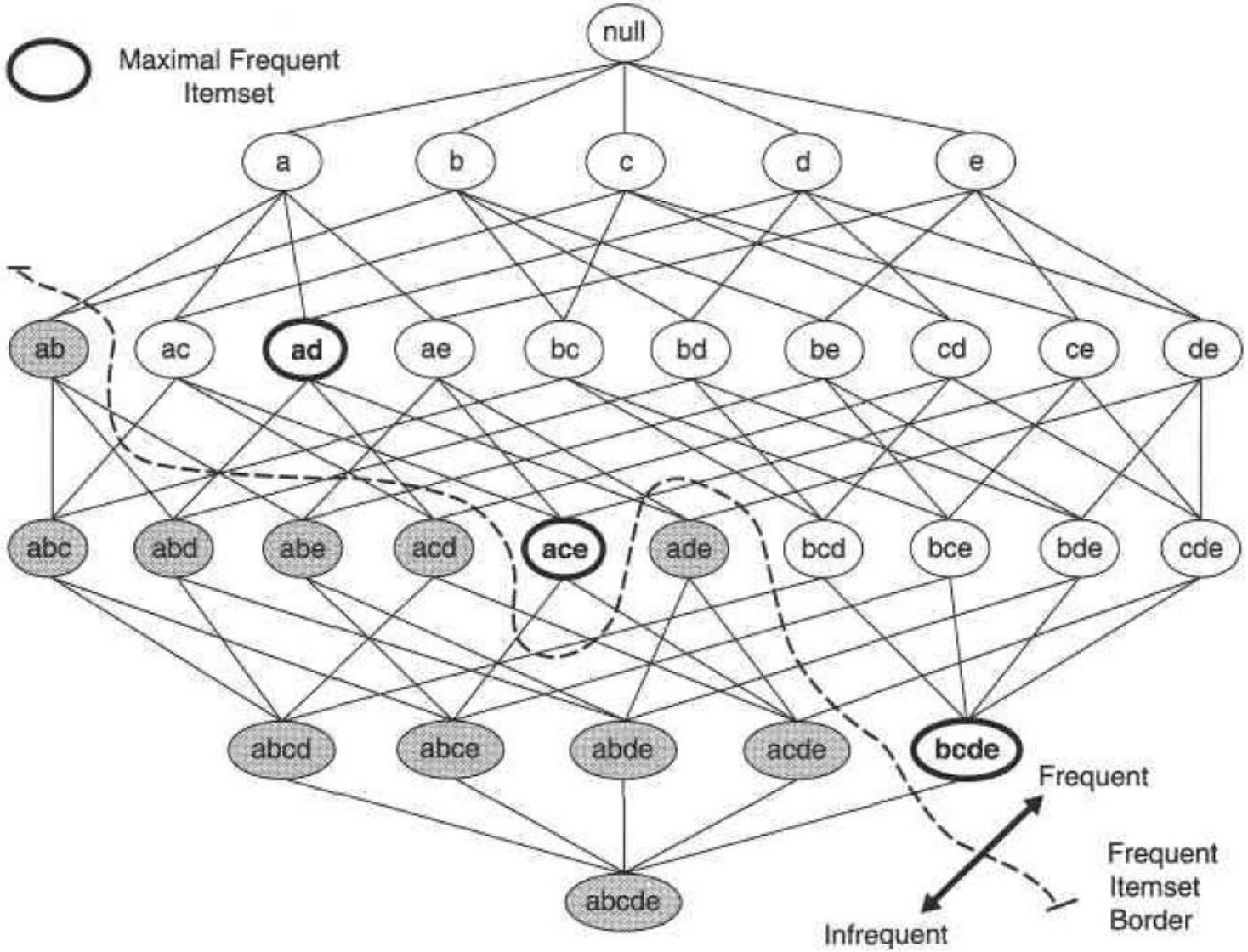
Each frequent k-itemset can produce up to 2^k-2 association rules (ignoring null transactions)

1. Confidence-based pruning
2. Rule generation in the Apriori algorithm



COMPACT REPRESENTATION OF FREQUENT ITEMSETS:

1. Maximal frequent itemsets
   None of its immediate superset are frequent
2. Closed frequent itemsets
   closed and its support is greater than min support (closed: None of its immediate superset has exactly same support count as of previous)

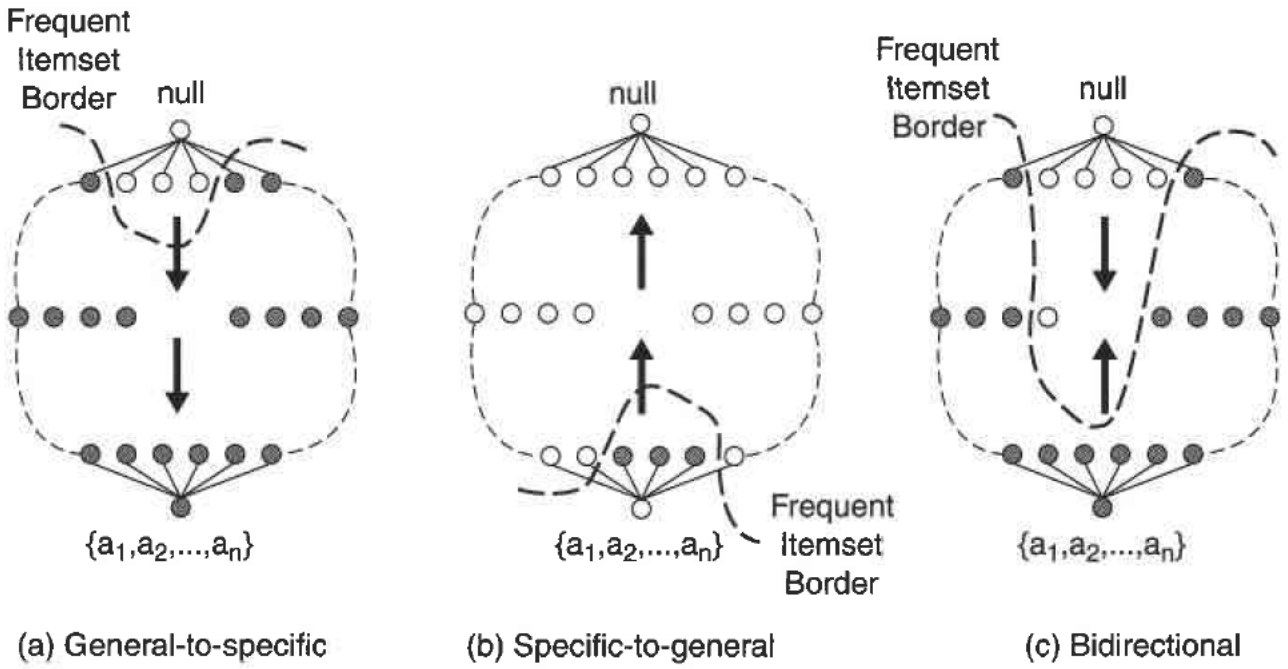


ALTERNATIVE METHOD FOR GENERATING FREQUENT ITEMSETS:

1. Traversal of itemset lattice:
   1. General-to-specific vs specific-to-general
   2. Equivalence classes
   3. Breadth-First vs Depth-First
2. Representation of transaction data set

ALTERNATIVE METHOD FOR GENERATING FREQUENT ITEMSETS:

1. Traversal of itemset lattice:
2. General-to-specific vs specific-to-general



FP-GROWTH:

1. It is a tree based algorithm.

2. It constructs conditional frequent pattern tree and conditional pattern base from database which satisfy minimum support.

3. FP Growth uses a depth-first search

4. FP Growth utilizes a pattern-growth approach means that, it only considers patterns actually existing in the database.

5. Runtime increases linearly, depending on the number of transactions and items

6. Data are very interdependent, each node needs the root.

7. It requires less memory space due to compact structure and no candidate generation.

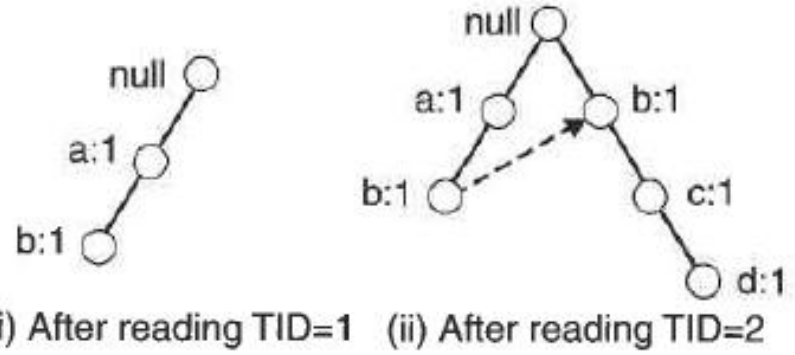
8. It scans the database only twice for constructing frequent pattern tree.

FP-GROWTH:

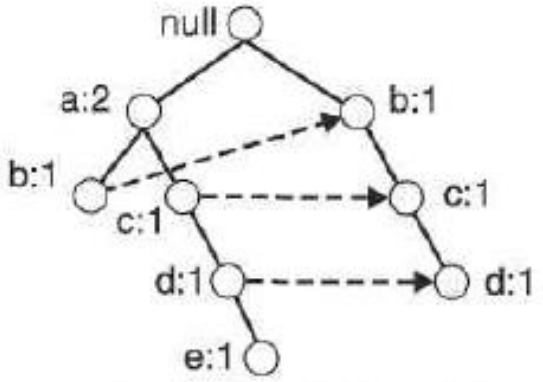
- Physical storage is higher because of pointers and counters for each node in data.
- What if we reverse the order?
- Divide-and-conquer (to find frequent itemset)
- Run-time depends on compaction factor

Transaction Data Set

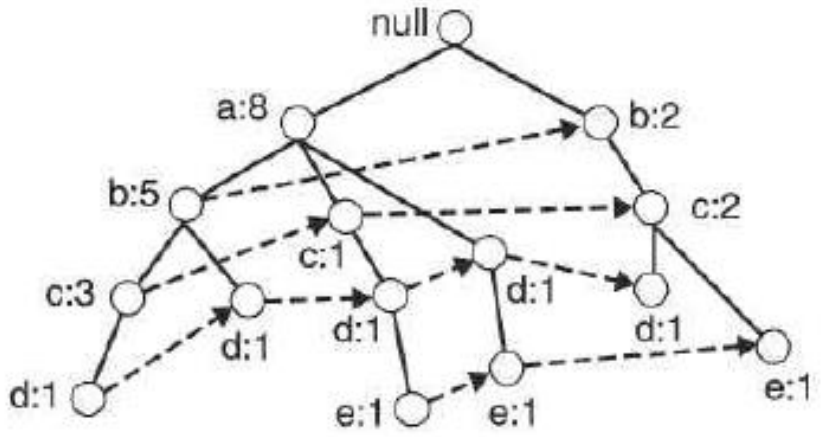
TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}



(i) After reading TID=1 (ii) After reading TID=2



(iii) After reading TID=3



(iv) After reading TID=10

FP-GROWTH:

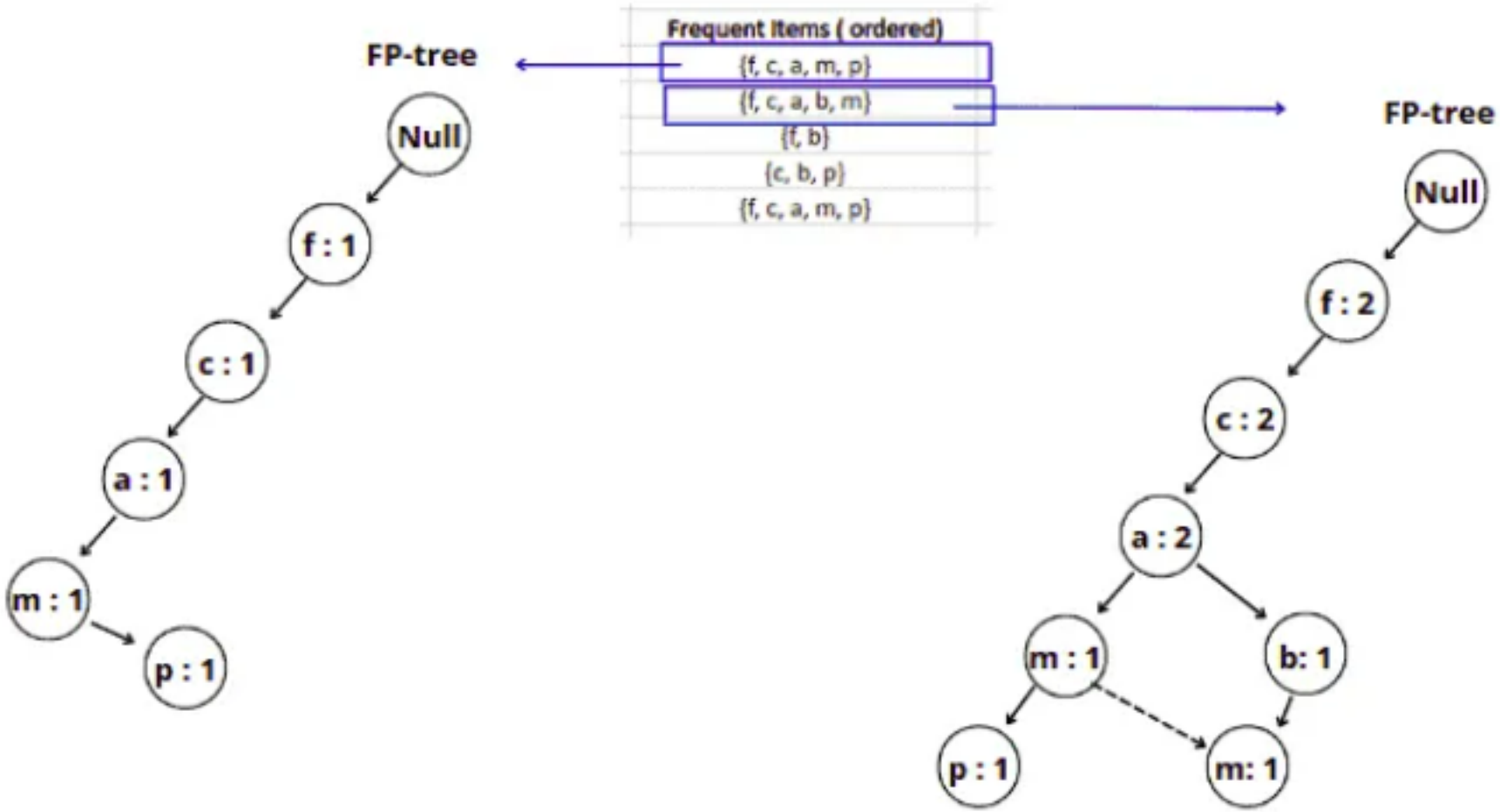
ID	Items bought
100	{f, a, c, d, g, i, m, p}
200	{a, b, c, f, l, m, o}
300	{b, f, h, j, o}
400	{b, c, k, s, p}
500	{a, f, c, e, l, p, m, n}

Item	Frequecy
{f}	4
{c}	3
{a}	3
{b}	3
{m}	3
{p}	3

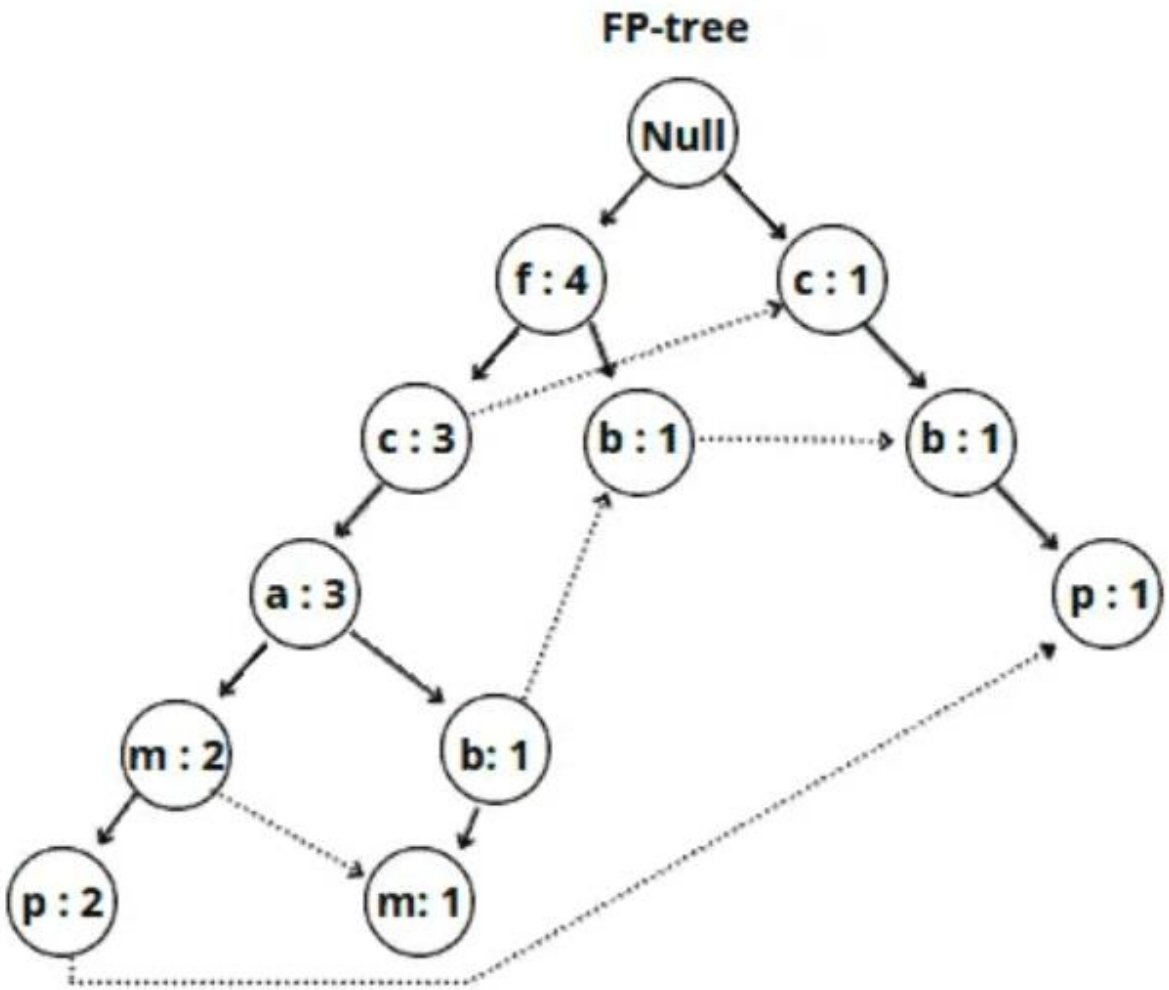
ID	Items bought
100	{f, a, c, d, g, i, m, p}
200	{a, b, c, f, l, m, o}
300	{b, f, h, j, o}
400	{b, c, k, s, p}
500	{a, f, c, e, l, p, m, n}

Frequent Items (ordered)
{f, c, a, m, p}
{f, c, a, b, m}
{f, b}
{c, b, p}
{f, c, a, m, p}

FP-GROWTH:


FP-tree
Null
f : 1
c : 1
a : 1
m : 1
p : 1
Frequent Items (ordered)
{f, c, a, m, p}
{f, c, a, b, m}
{f, b}
{c, b, p}
{f, c, a, m, p}
FP-tree
Null
f : 2
c : 2
a : 2
m : 1
b: 1
p : 1
m: 1

FP-GROWTH:



Item	Conditional Pattern Base
p	{{ f, c, a, m : 2}, { c, b : 1}}
m	{{ f, c, a : 2}, {f, c, a, b : 1}}
b	{{f :1}, { c:1}, {f, c, a : 1}}
a	{{f, c : 3}}
c	{{ f : 3}}

Item	Conditional Pattern Base	Conditional FP-tree
p	{{f, c, a, m : 2 }, {c, b : 1 }}	{c:3 }
m	{{f, c, a : 2}, {f, c, a, b : 1 }}	{f:3, c:3, a:3 }
b	{{f : 1 }, {c : 1}, {f, c, a : 1 }}	--
a	{{f, c : 3 }}	{f:3, c:3 }
c	{{f : 3 }}	{f:3 }

FP-GROWTH:

Item	Conditional Pattern Base	Conditional FP-tree	Generated Frequent Patterns
p	{{f, c, a, m : 2 }, {c, b : 1 }}	{c:3 }	{c, p : 3 }
m	{{f, c, a : 2}, {f, c, a, b : 1 }}	{f:3, c:3, a:3 }	{f, m : 3 }, {c, p : 3 }, {a, m : 3 }, {f, c, m : 3}, {f, a, m : 3}, {c, a, m : 3 }, {f, c, a, m : 3 }
b	{{f : 1 }, {c : 1}, {f, c, a : 1 }}	--	--
a	{{f, c : 3 }}	{f:3, c:3 }	{f, a : 3 }, {c, a : 3 }, {f, c, a : 3 }
c	{{f : 3 }}	{f:3 }	{f, c : 3 }

APRIORI VS FP GROWTH:

	FP Growth	Apriori
Speed	Faster, runtime increases linearly with increase in number of itemsets	Slower, runtime increases exponentially with increase in number of itemsets
Memory	Small, storing the compact version of database	Large, all the candidates from self-joining are stored in the memory
Candidates	No candidate generation	Use self-joining for candidate generation
Frequent patterns	Pattern growth achieved by mining conditional FP trees.	Patterns selected from the candidates whose support is higher than minSup.
Scans	Only require two scans	Scan the database over and over again.

EVALUATION OF ASSOCIATION PATTERN:

1. Objective interestingness measure
2. Measures beyond pairs of binary variables: **Simpson's Paradox**
3. Subjective interestingness measure:
    1. Based on domain information such as concept hierarchy or profit margin of items
    2. Used to filter patterns that are obvious and non-actionable.
    3. Visualization
    4. Template-based approach: *This approach allows the users to constrain the type of patterns extracted by the mining algorithm. Instead of reporting all the extracted rules, only rules that satisfy a user-specified template are returned to the users.*

EVALUATION OF ASSOCIATION PATTERN:

1. Objective interestingness measure
   1. Support, confidence, and correlation
   2. Limitations of the support-confidence framework:
      1. Interesting pattern with low support will be eliminated by the support threshold
      2. Confidence measure ignores the support of the itemset in the rule consequent
      3. Above point can be enhanced by using **LIFT** parameter: **which computes the ratio between the rule's confidence and the support of the itemset in the rule consequent**

$$Lift = \frac{c(A \longrightarrow B)}{s(B)},$$

OBJECTIVE INTERESTINGNESS MEASURE:

Interest Factor: equivalent to Lift factor

$$I(A,B) = \frac{s(A,B)}{s(A) \times s(B)} = \frac{N f_{11}}{f_{1+} f_{+1}}.$$

	B	$\overline{B}$	
A	f_{11}	f_{10}	f_{1+}
$\overline{A}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	N

Fig: Contingency Table

Limitation of Interest Factor: not accurate for all possible association in text.

Correlation Analysis: Statistical based approach to analyze relation between items

For continuous variables: Pearson's correlation coefficient,

For binary variables: Ø-coefficient

$$\phi = \frac{f_{11} f_{00} - f_{01} f_{10}}{\sqrt{f_{1+} f_{+1} f_{0+} f_{+0}}}.$$

Limitation of correlation analysis: 1) equal importance to co-presence and co-absence.

2) it does not remain invariant when there are proportional changes to the sample size

OBJECTIVE INTERESTINGNESS MEASURE:

	B	$\overline{B}$	
A	f_{11}	f_{10}	f_{1+}
$\overline{A}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	N

Fig: Contingency Table

IS Measure: for handling asymmetric binary variables

$$IS(A,B) = \sqrt{I(A,B) \times s(A,B)} = \frac{s(A,B)}{\sqrt{s(A)s(B)}}.$$

Large when support is large.

Contrary to correlation analysis.

Similar to cosine measure:

$$IS(A,B) = \frac{s(A,B)}{\sqrt{s(A) \times s(B)}} = \frac{\mathbf{A} \bullet \mathbf{B}}{|\mathbf{A}| \times |\mathbf{B}|} = cosine(\mathbf{A},\mathbf{B}).$$

Limitation of IS Measure: Value can be large even for uncorrelated or negatively correlated documents.

SYMMETRIC OBJECTIVE MEASURE :

Table 6.11. Examples of symmetric objective measures for the itemset $\{A, B\}$.

Measure (Symbol)	Definition
Correlation (ϕ)	$\frac{Nf_{11}-f_{1+}f_{+1}}{\sqrt{f_{1+}f_{+1}f_{0+}f_{+0}}}$
Odds ratio (α)	$(f_{11}f_{00})/(f_{10}f_{01})$
Kappa (κ)	$\frac{Nf_{11}+Nf_{00}-f_{1+}f_{+1}-f_{0+}f_{+0}}{N^2-f_{1+}f_{+1}-f_{0+}f_{+0}}$
Interest (I)	$(Nf_{11})/(f_{1+}f_{+1})$
Cosine (IS)	$(f_{11})/(\sqrt{f_{1+}f_{+1}})$
Piatetsky-Shapiro (PS)	$\frac{f_{11}}{N} - \frac{f_{1+}f_{+1}}{N^2}$
Collective strength (S)	$\frac{f_{11}+f_{00}}{f_{1+}f_{+1}+f_{0+}f_{+0}} \times \frac{N-f_{1+}f_{+1}-f_{0+}f_{+0}}{N-f_{11}-f_{00}}$
Jaccard (ζ)	$f_{11}/(f_{1+}+f_{+1}-f_{11})$
All-confidence (h)	$\min\left[\frac{f_{11}}{f_{1+}}, \frac{f_{11}}{f_{+1}}\right]$

ASYMMETRIC OBJECTIVE MEASURE :

Table 6.12. Examples of asymmetric objective measures for the rule $A \longrightarrow B$.

Measure (Symbol)	Definition
Goodman-Kruskal (λ)	$\left(\sum_j \max_k f_{jk} - max_k f_{+k}\right) / \left(N - \max_k f_{+k}\right)$
Mutual Information (M)	$\left(\sum_i \sum_j \frac{f_{ij}}{N} \log \frac{N f_{ij}}{f_{i+} f_{+j}}\right) / \left(-\sum_i \frac{f_{i+}}{N} \log \frac{f_{i+}}{N}\right)$
J-Measure (J)	$\frac{f_{11}}{N} \log \frac{N f_{11}}{f_{1+} f_{+1}} + \frac{f_{10}}{N} \log \frac{N f_{10}}{f_{1+} f_{+0}}$
Gini index (G)	$\frac{f_{1+}}{N} \times \left(\frac{f_{11}}{f_{1+}}\right)^2 + \left(\frac{f_{10}}{f_{1+}}\right)^2] - \left(\frac{f_{+1}}{N}\right)^2$ $+ \frac{f_{0+}}{N} \times [\left(\frac{f_{01}}{f_{0+}}\right)^2 + \left(\frac{f_{00}}{f_{0+}}\right)^2] - \left(\frac{f_{+0}}{N}\right)^2$
Laplace (L)	$\left(f_{11} + 1\right) / \left(f_{1+} + 2\right)$
Conviction (V)	$\left(f_{1+} f_{+0}\right) / \left(N f_{10}\right)$
Certainty factor (F)	$\left(\frac{f_{11}}{f_{1+}} - \frac{f_{+1}}{N}\right) / \left(1 - \frac{f_{+1}}{N}\right)$
Added Value (AV)	$\frac{f_{11}}{f_{1+}} - \frac{f_{+1}}{N}$

PROPERTIES OF OBJECTIVE MEASURES :

1. Inversion property:
   1. An objective measure M is invariant under the inversion operation if its value remains the same when exchanging the frequency counts f_{11} with f_{00} and f_{10} with f_{01}
2. Scaling property:
   1. An objective measure M is invariant under the row/column scaling operation
3. Null addition property:
   1. An objective measure M is invariant under the null addition operation if it is not affected by increasing f_{00}, while all other frequencies in the contingency table stay the same.

SIMPSON'S PARADOX :

1. Measures beyond pairs of binary variables: **Simpson's Paradox**
2. Interpreting the association between variables because the observed relationship may be influenced by the presence of other confounding factors, i.e., hidden variables that are not included in the analysis.
3. In some cases, the hidden variables may cause the observed relationship between a pair of variables to disappear or reverse its direction, a phenomenon that is known as **Simpson's paradox.**

SIMPSON'S PARADOX:

{HDTV:Yes} → {Exercise machine:Yes}, confidence of 99/180 =55%

{HDTV:No} → {Exercise machine:Yes}, confidence of 54/120 =45%

Rules suggest that customers who buy: high-definition televisions are more likely to buy exercise machines than those who do not buy high-definition televisions

Depend on the category of people.

For College Students:

{HDTV:Yes} → {Exercise machine:Yes}, confidence of 1/10 =10%

{HDTV:No} → {Exercise machine:Yes}, confidence of 4/34 =11.8%

For Working Adults:

{HDTV:Yes} → {Exercise machine:Yes}, confidence of 98/170 =57.7%

{HDTV:No} → {Exercise machine:Yes}, confidence of 50/86 =58.1%

Positively correlated in combined data and negatively correlated in stratified data. (Reverse in direction of association is Simpson,s paradox.

EFFECT OF SKEWED SUPPORT DISTRIBUTION:

1. Choosing right support threshold is tricky

Table 6.21. Grouping the items in the census data set based on their support values.

Group	G_1	G_2	G_3
Support	$< 1\%$	$1\% - 90\%$	$> 90\%$
Number of Items	1735	358	20

2. Threshold =20% may miss interesting associations
3. If a threshold is too low:
   1. Computational and memory requirements will increase considerably
   2. Number of extracted patterns will increase
   3. May extract many spurious patterns (high-frequency item with low frequency)
      They are cross-support and posses weak correlation

EFFECT OF SKEWED SUPPORT DISTRIBUTION:

1. Before eliminating such patterns we need cross-support patterns.
2. Cross support pattern is an itemset X = {i_1, i_2, …. I_k) whose support ratio is less than a user-specified threshold h_c

$$r(X) = \frac{\min\left[s(i_1), s(i_2), \ldots, s(i_k)\right]}{\max\left[s(i_1), s(i_2), \ldots, s(i_k)\right]},$$

3. Support for milk is 70%, sugar is 10% and caviar is 0.04%

$$r = \frac{\min\left[0.7, 0.1, 0.0004\right]}{\max\left[0.7, 0.1, 0.0004\right]} = \frac{0.0004}{0.7} = 0.00058 < 0.01.$$

EFFECT OF SKEWED SUPPORT DISTRIBUTION:

1. Difficulty in extracting data from cross-support and non-cross-support patterns.
2. Cross-support patterns can be detected by examining the lowest confidence rule.

PROOF FOR CROSS-SUPPORT:

1. Anti-monotone property of confidence:
   1. This property suggests that confidence never increases as we shift more items from the left- to the right-hand side of an association rule.
2. Frequent itemset rule: Rule will have lowest confidence if

$$s(i_j) = \max\left[s(i_1), s(i_2), \ldots, s(i_k)\right]$$

3. Lowest confidence attainable from frequent itemset is:
   1. H-confidence must not exceed than
   2. H-confidence of the pattern is auaranteed to be less than threshold

$$\text{h-confidence}(X) \leq \frac{\min\left[s(i_1), s(i_2), \ldots, s(i_k)\right]}{\max\left[s(i_1), s(i_2), \ldots, s(i_k)\right]}.$$

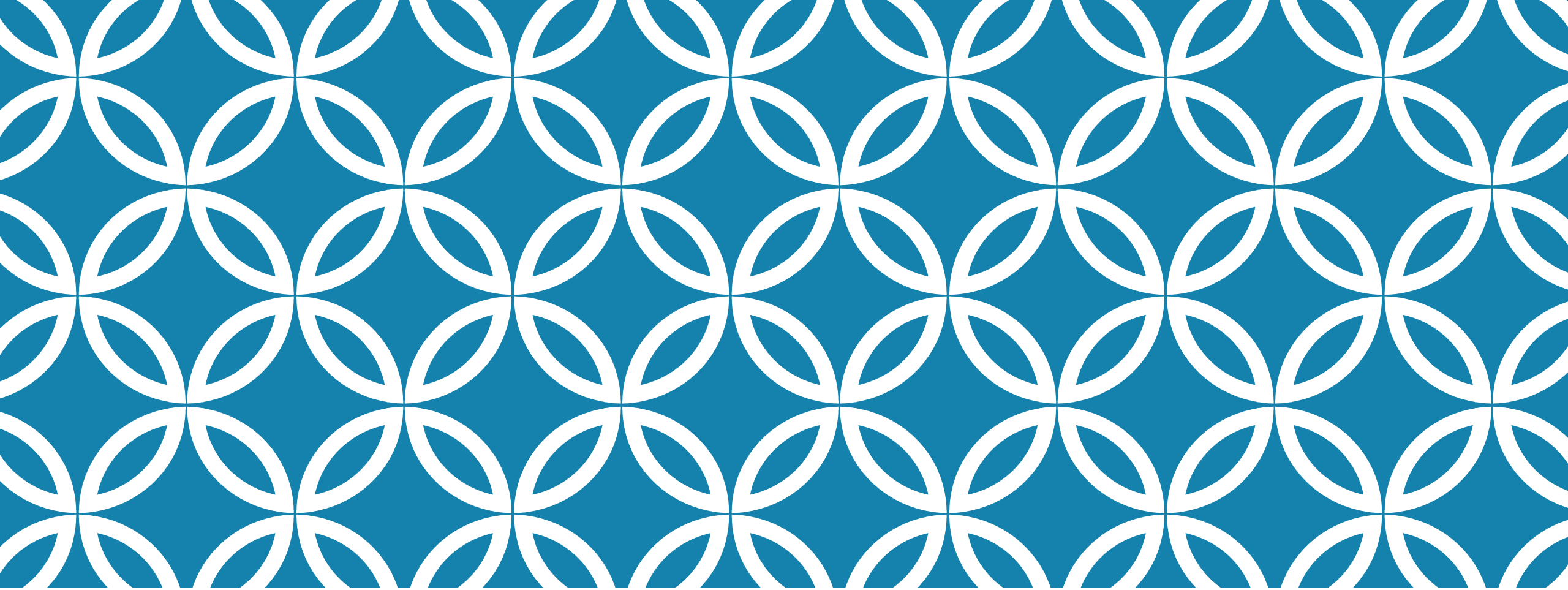
p	q	r
0	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
0	0	0
0	0	0
0	0	0
0	0	0

CROSS-SUPPORT :

1. Cross-support patterns can be eliminated by ensuring that the h-confidence values for the patterns exceed h_c
2. Using h-confidence go beyond eliminating cross-support patterns.
3. Can be incorporated directly into data mining algorithm.
4. H-confidence ensures that the items contained in a itemset are strongly associated with each other.
5. Such strongly associated patterns are called **hyperclique patterns.**

THANK YOU..

Here it ends with unit 3.



UNIT – 4
UNSUPERVISED LEARNING & CLUSTERING

By Rashmi Bhattad

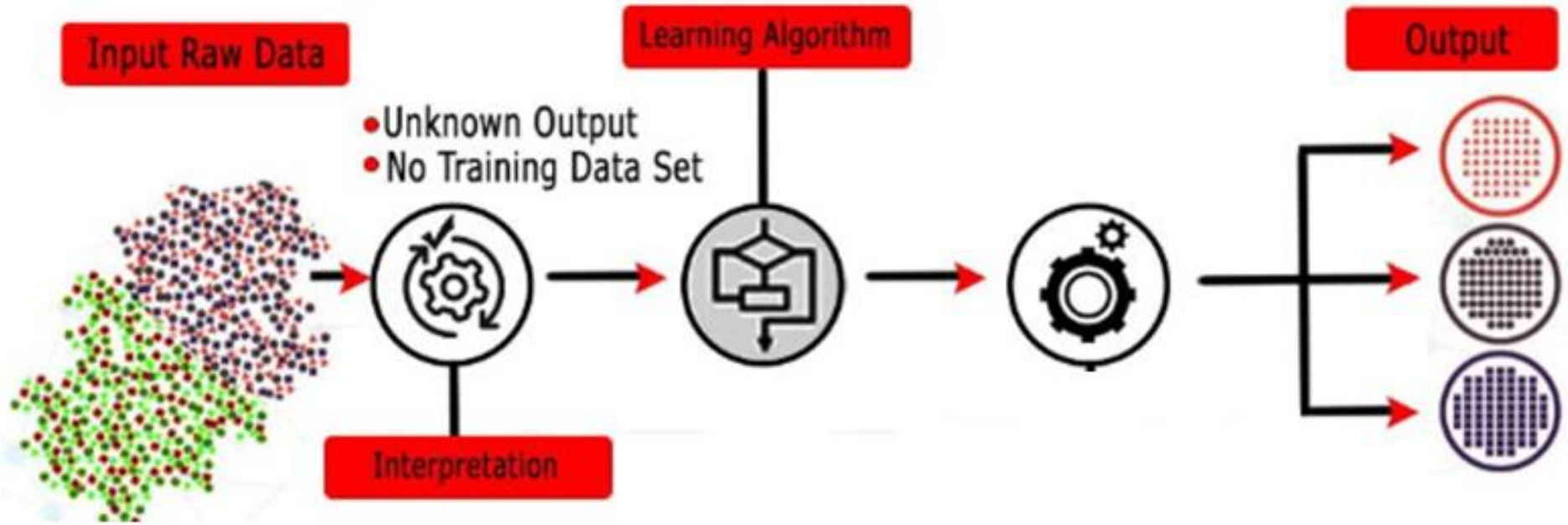
CLUSTERING:

- Clustering is an unsupervised machine learning technique.
- It is the process of division of the dataset into groups in which the members in the same group possess similarities in features.
- The commonly used clustering algorithms are:
  - Centroid Based clustering: K-Means
  - Hierarchical clustering: Agglomerative
  - Density-based clustering: DBSCAN
  - Model-based/Distribution clustering: EM Algorithm, soft boundaries, etc.

CLUSTERING:

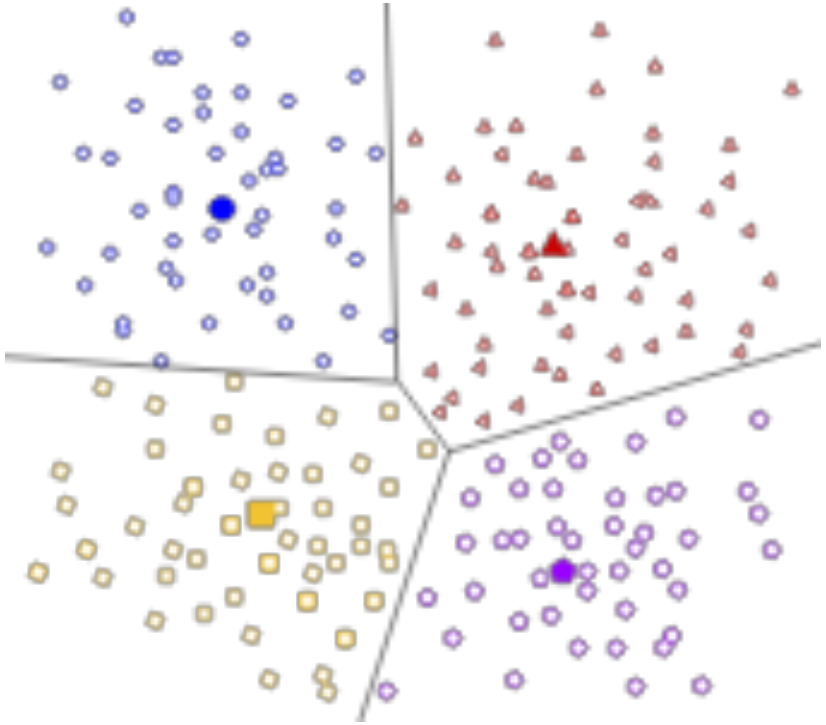
- Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.
- In clustering, we do not have a target to predict. We look at the data and then try to club similar observations and form different groups.
- All the data points in a cluster should be similar to each other.
- The data points from different clusters should be as different as possible.

CLUSTERING:


Input Raw Data
Learning Algorithm
Output
Unknown Output
No Training Data Set
Interpretation

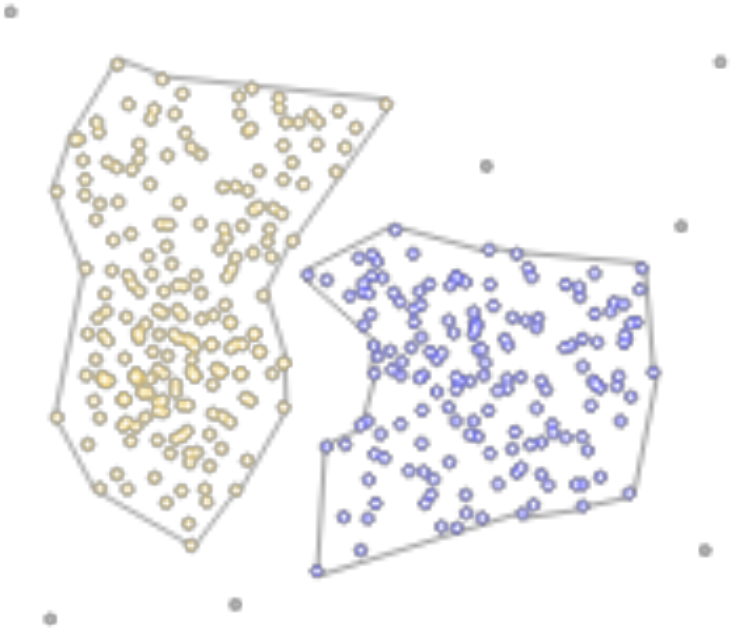
CENTROID-BASED CLUSTERING:

Centroid-based clustering organizes the data into non-hierarchical clusters, in contrast to hierarchical clustering defined below. k-means is the most widely-used centroid-based clustering algorithm. Centroid-based algorithms are efficient but sensitive to initial conditions and outliers. This course focuses on k-means because it is an efficient, effective, and simple clustering algorithm.



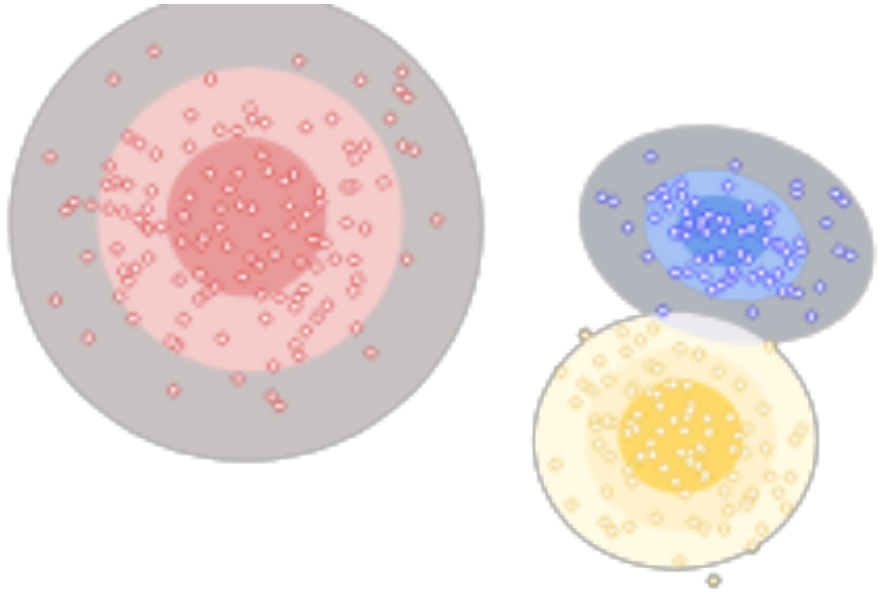
DENSITY-BASED CLUSTERING:

Density-based clustering connects areas of high example density into clusters. This allows for arbitrary-shaped distributions as long as dense areas can be connected. These algorithms have difficulty with data of varying densities and high dimensions. Further, by design, these algorithms do not assign outliers to clusters.



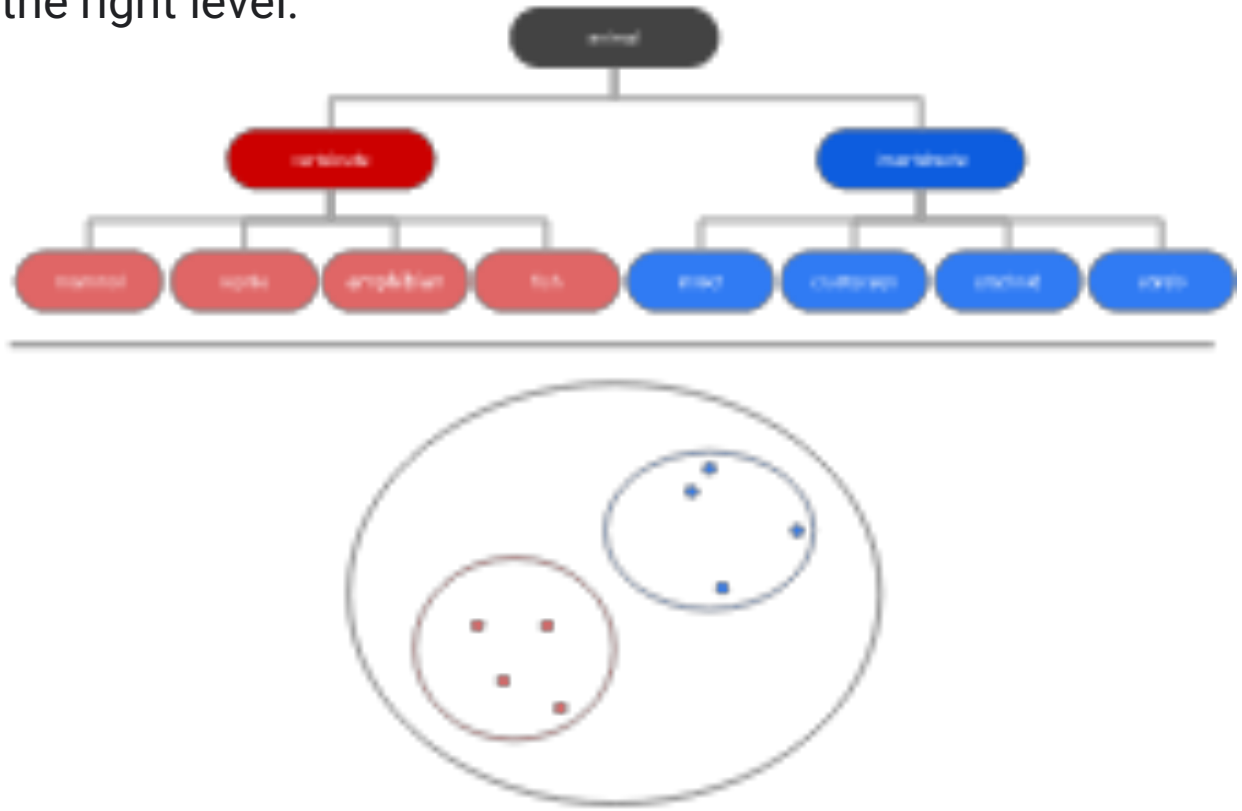
DISTRIBUTION-BASED CLUSTERING:

This clustering approach assumes data is composed of distributions, such as Gaussian distributions. The distribution-based algorithm clusters data into three Gaussian distributions. As distance from the distribution's center increases, the probability that a point belongs to the distribution decreases. The bands show a decrease in probability. When you do not know the type of distribution in your data, you should use a different algorithm.

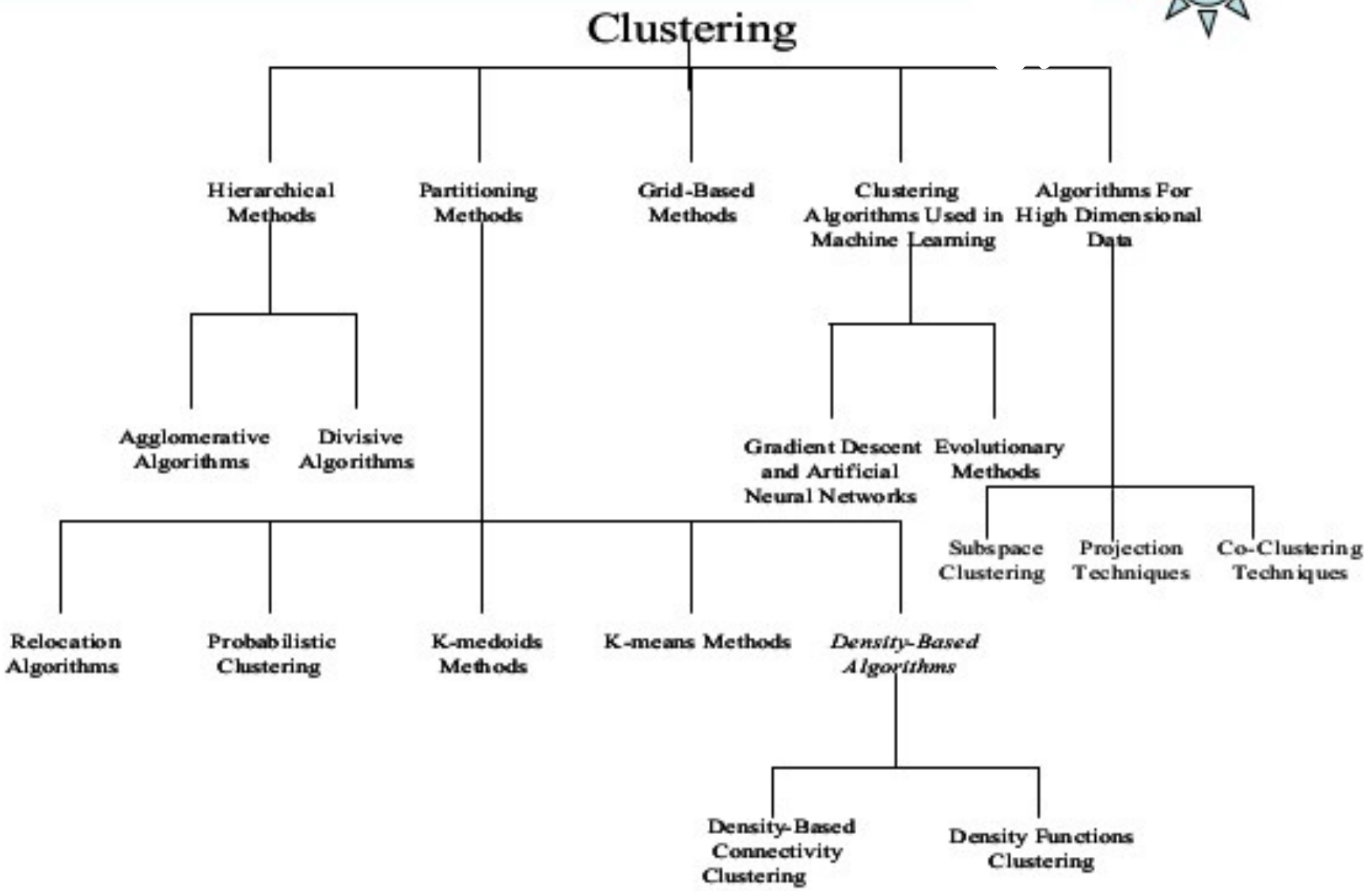


HIERARCHICAL CLUSTERING:

Hierarchical clustering creates a tree of clusters. Hierarchical clustering, not surprisingly, is well suited to hierarchical data, such as taxonomies. In addition, another advantage is that any number of clusters can be chosen by cutting the tree at the right level.



Types of Clustering Algorithms



Source: https://medium.com/@pytyagi/clustering-83f210299e47

DISTANCE METRIC:

To measure the similarity between continuous or numerical variables

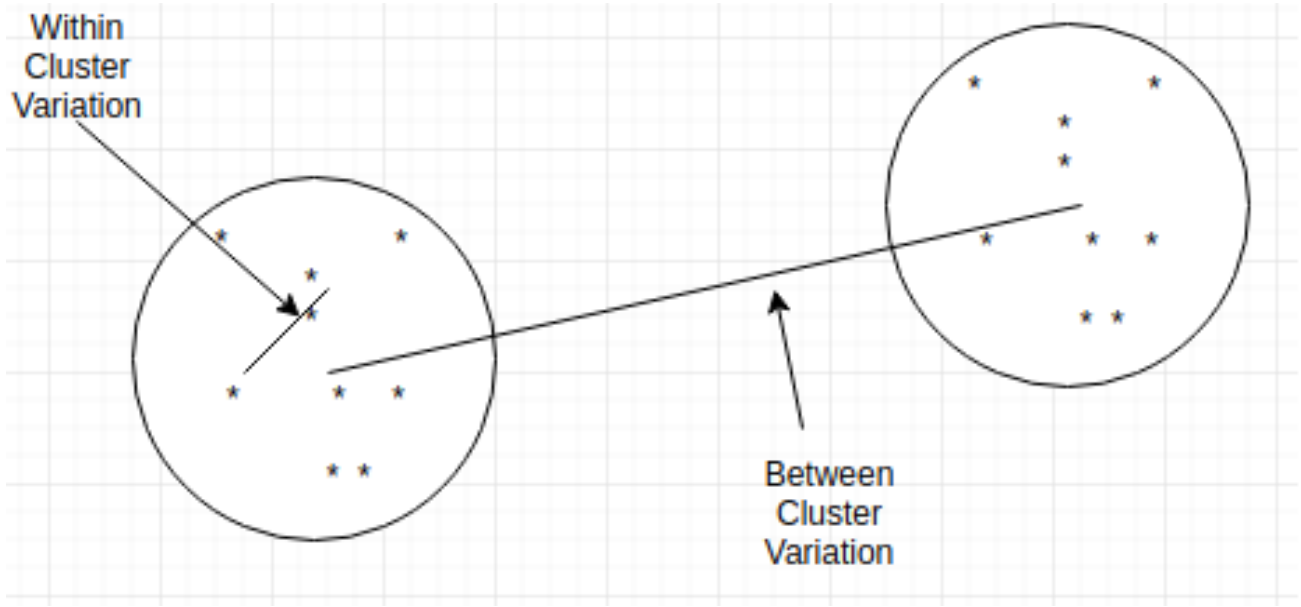
1. Euclidean Distance
2. Manhattan Distance: Manhattan distance is usually preferred over the more common Euclidean distance when there is high dimensionality in the data.
3. Minkowski Distance
4. To measure similarity between categorical variables: Hamming Distance *(Hamming distance is used to measure the distance between categorical variables.)*
5. To measure similarities between different documents: Cosine distance *(Cosine distance metric is mainly used to find the amount of similarity between two data points.)*

CLUSTERING EVALUATION METRIC:

1. Clusters are mainly evaluated with either of these two approaches
2. Internal measures:
    1. Intra-cluster distance: where the sum of distances between objects in the same cluster are minimized,
    2. Inter-cluster distance: where the distances between different clusters are maximized.
3. External measures:
    1. are related to how representative are the current clusters to “true” classes.
    2. Measured in terms of purity, entropy or F-measure

CLUSTERING EVALUATION METRIC:

1. WCV (Within Cluster Variation): The variation in the data points that are present in the cluster.
2. BCV (Between Cluster Variation): The variation between 2 clusters.



UNSUPERVISED CLUSTERING VALIDATION:

In the case of unsupervised clustering to validate the performance following matrices are used:

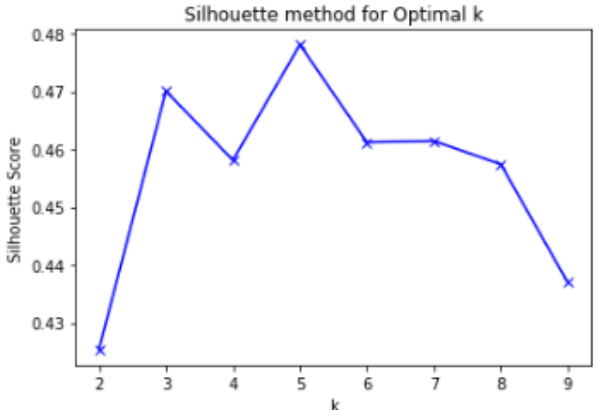
1. Cluster Cohesion and cluster separation
2. Sum square error (SSE)
3. Silhouette Coefficient
4. Internal indices

SILHOUETTE ALGORITHM :

1. The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).
2. The silhouette ranges from −1 to +1, where a high value indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters.
3. A value close to 1 implies that the instance is close to its cluster and is a part of the right cluster. Whereas, a value close to -1 means that the value is assigned to the wrong cluster. Whereas 0 indicates the point is on the boundary.
4. If most objects have a high value, then the clustering configuration is appropriate.
5. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.
6. This is calculated for each point in the dataset

SILHOUETTE ALGORITHM :

- This method is better as it makes the decision regarding the optimal number of clusters more meaningful and clear. But this metric is computation expensive as the coefficient is calculated for every instance.
- Silhouette's coefficient is calculated by: : $$s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$$
- *where y is the mean intra-cluster distance: mean distance to the other instances in the same cluster. X depicts the mean nearest cluster distance i.e. mean distance to the instances of the next closest cluster.*



- k=5 should be chosen for the number of clusters.

K-MEANS:

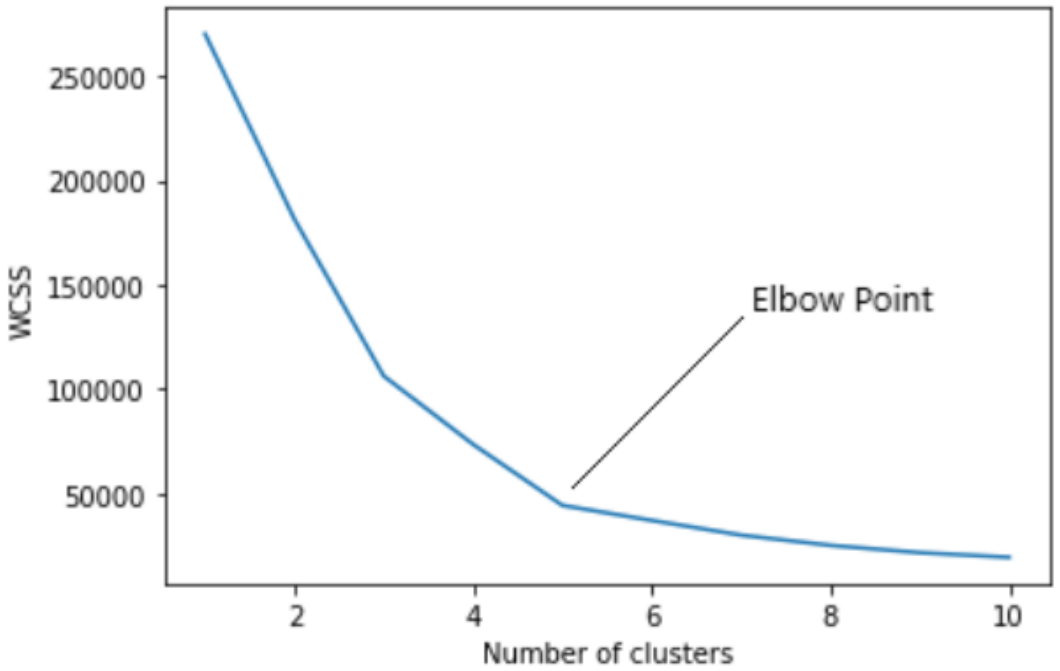
1. Select the number of clusters for the dataset (K)
2. Select K number of centroids
3. By calculating the Euclidean distance or Manhattan distance assign the points to the nearest centroid, thus creating K groups
4. Now find the original centroid in each group
5. Again reassign the whole data point based on this new centroid, then repeat step 4 until the position of the centroid doesn't change and all data points remain in the same cluster.

Finding the optimal number of clusters is an important part of this algorithm. A commonly used method for finding optimal K value is Elbow Method

ELBOW METHOD:

- In the Elbow method, we are actually varying the number of clusters (K) from 1 – 10.
- For each value of K, we are calculating WCSS (Within-Cluster Sum of Square).
- WCSS is the sum of squared distance between each point and the centroid in a cluster.
- When we plot the WCSS with the K value, the plot looks like an Elbow. As the number of clusters increases, the WCSS value will start to decrease.
- WCSS value is largest when K = 1. When we analyze the graph we can see that the graph will rapidly change at a point and thus creating an elbow shape.
- From this point, the graph starts to move almost parallel to the X-axis. The K value corresponding to this point is the optimal K value or an optimal number of clusters.

ELBOW METHOD:



KNN EXAMPLE:

X1 = Acid Durability	X2 = Strength	Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Test for X1 = 3 and X2 = 7 (Use k = 3)

X1 = Acid Durability	X2 = Strength	Square distance to query instance
7	7	$(7-3)^2 + (7-7)^2 = 16$
7	4	$(7-3)^2 + (4-7)^2 = 25$
3	4	$(3-3)^2 + (4-7)^2 = 9$
1	4	$(1-3)^2 + (4-7)^2 = 13$

KNN EXAMPLE:

Test for X1 = 3 and X2 = 7 (Use k = 3)

X1 = Acid Durability	X2 = Strength	Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

X1 = Acid Durability	X2 = Strength	Square distance to query instance	Rank Min Dist	Is it included in 3-NN
7	7	$(7-3)^2 + (7-7)^2 = 16$	3	Yes
7	4	$(7-3)^2 + (4-7)^2 = 25$	4	No
3	4	$(3-3)^2 + (4-7)^2 = 9$	1	Yes
1	4	$(1-3)^2 + (4-7)^2 = 13$	2	Yes

Final category of instance: Good

SILHOUETTE ON KMEANS :

1. Following is the given data, calculate the silhouette metric for following cluster data. Considering (1,0) as point in question.

```
Cluster 1 ={{1,0},{1,1}}
Cluster 2 ={{1,2},{2,3},{2,2},{1,2}},
Cluster 3 ={{3,1},{3,3},{2,1}}
```

2. Distance of point w.r.t cluster 1: `a1 =√( (1-1)^2 + (0-1)^2) =√(0+1)=√1=1`
3. Distance of point w.r.t cluster 2: Average is 2.325

```
{1,0} ----> {1,2} = distance = √((1-1)^2 + (0-2)^2) =√(0+4)=√4=2
{1,0} ----> {2,3} = distance = √((1-2)^2 + (0-3)^2) =√(1+9)=√10=3.16
{1,0} ----> {2,2} = distance = √((1-2)^2 + (0-2)^2) =√(1+4)=√5=2.24
{1,0} ----> {1,2} = distance = √((1-1)^2 + (0-2)^2) =√(0+4)=√4=2
```

SILHOUETTE ON KMEANS :

1. Distance of point w.r.t cluster 3: Average is 2.4

```
{1,0} ----> {3,1} = distance = √((1-3)^2 + (0-1)^2) =√(4+1)=√5=2.24
{1,0} ----> {3,3} = distance = √((1-3)^2 + (0-3)^2) =√(4+9)=√13=3.61
{1,0} ----> {2,1} = distance = √((1-2)^2 + (0-1)^2) =√(1+1)=√2=1.41
```

2. So the silhouette coefficient of cluster 1 is as follows:

```
s1= 1-(a1/b1) = 1- (1/2.325)=1-0.4301=0.5699
```

LABELLED VS UNLABELLED DATA:

Labelled Data: Play Tennis

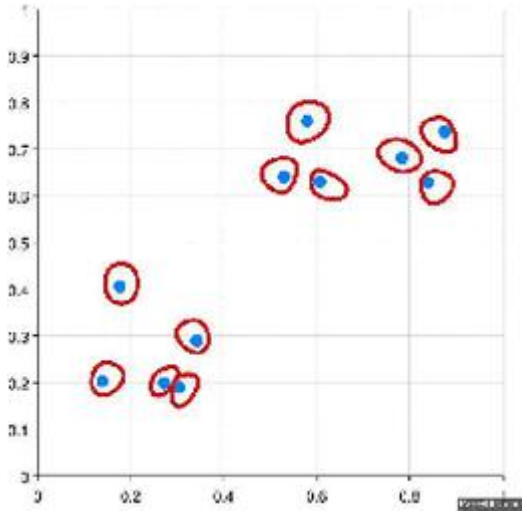
Unlabelled Data: Mall Customers

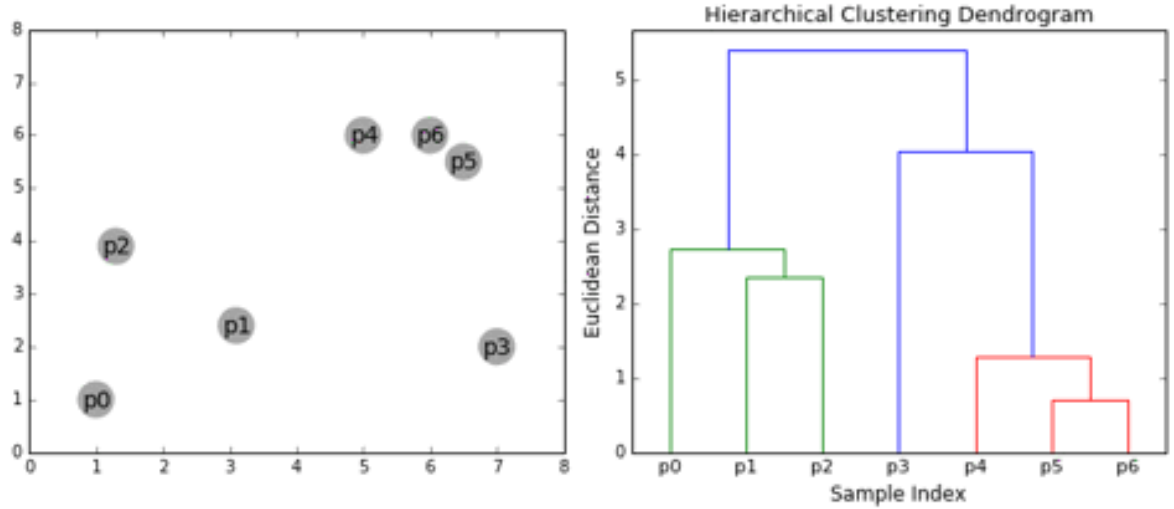
Index	ustomerI[	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

AGGLOMERATIVE CLUSTERING:

- Agglomerative: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- We assign each point to an individual cluster in this technique. Then, at each iteration, we merge the closest pair of clusters and repeat this step until only a single cluster is left.
- In this approach we are merging (or adding) the clusters at each step. So it is also known as additive hierarchical clustering.
- This clustering algorithm does not require us to prespecify the number of clusters.

AGGLOMERATIVE CLUSTERING:





https://www.kdnuggets.com/2019/09/hierarchical-clustering.html

AGGLOMERATIVE CLUSTERING EXAMPLE:

Sample No.	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

- Step 1: Compute Euclidean distance between all points in the given list. Distance between p1 to p2 is given as follows:

$$d(P1, P2) = \sqrt{(0.4 - 0.22)^2 + (0.53 - 0.38)^2} = \sqrt{(0.18)^2 + (0.15)^2} = \sqrt{0.0324 + 0.0225} = 0.23$$

- By calculating this for all the variables, final matrix will look like:

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.23 & 0 & & & & \\ P3 & 0.22 & 0.14 & 0 & & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 & \\ P6 & 0.24 & 0.24 & 0.10 & 0.22 & 0.39 & 0 \end{pmatrix}$$

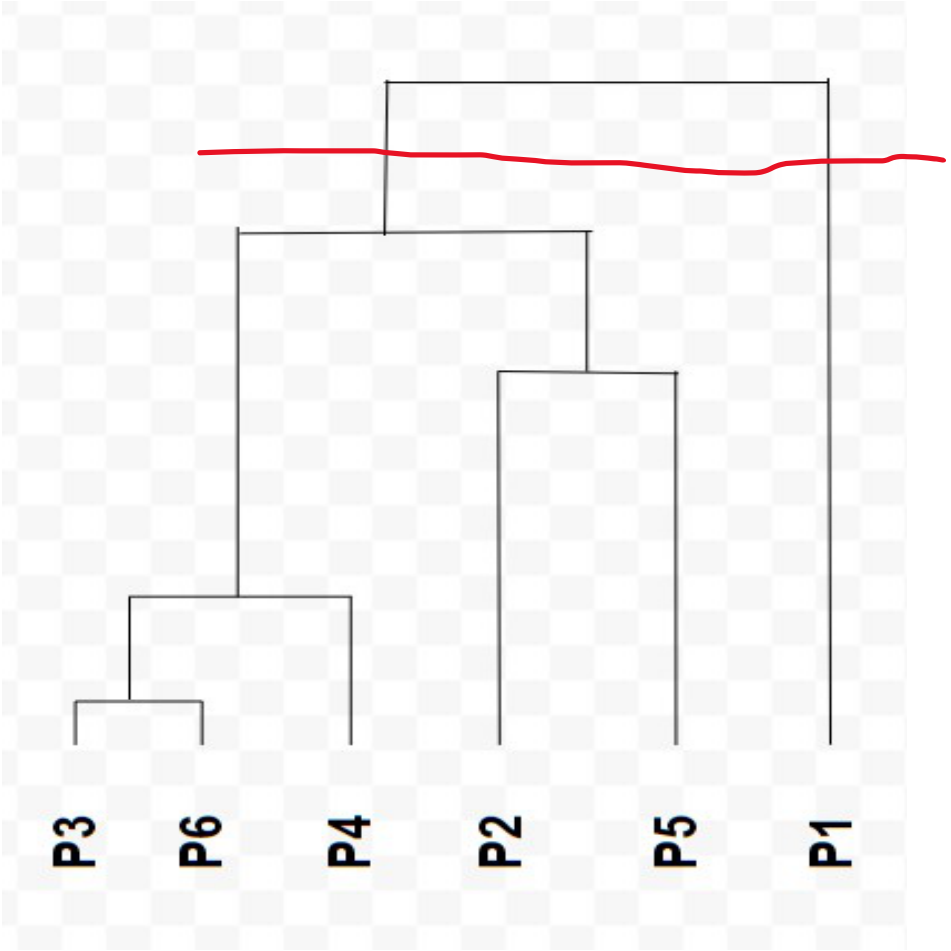
AGGLOMERATIVE CLUSTERING EXAMPLE:

Step 1: Check the minimum distance first

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.23 & 0 & & & & \\ P3 & 0.22 & 0.14 & 0 & & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 & \\ P6 & 0.24 & 0.24 & 0.10 & 0.22 & 0.39 & 0 \end{pmatrix} \begin{pmatrix} & P1 & P2 & P3, P6 & P4 & P5 \\ P1 & 0 & & & & \\ P2 & 0.23 & 0 & & & \\ P3, P6 & 0.22 & 0.14 & 0 & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 \end{pmatrix} \begin{pmatrix} & P1 & P2 & P3, P6, P4 & P5 \\ P1 & 0 & & & \\ P2 & 0.23 & 0 & & \\ P3, P6, P4 & 0.22 & 0.14 & 0 & \\ P5 & 0.34 & 0.14 & 0.23 & 0 \end{pmatrix}$$

$$\begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix} \begin{pmatrix} & P1 & P2, P5, P3, P6, P4 \\ P1 & 0 & \\ P2, P5, P3, P6, P4 & 0.22 & 0 \end{pmatrix}$$

NUMBER OF CLUSTERS IN HIERARCHICAL CLUSTERING:



WEAKNESS OF HIERARCHICAL CLUSTERING:

- The weaknesses are that it rarely provides the best solution, it involves lots of arbitrary decisions
- It does not work with missing data
- It works poorly with mixed data types
- It does not work well on very large data sets
- Its main output, the dendrogram, is commonly misinterpreted.
- With many types of data, it is difficult to determine how to compute a distance matrix. There is no straightforward formula that can compute a distance where the variables are both numeric and qualitative.
  - For example, how can one compute the distance between a 45-year-old man, a 10-year-old-girl, and a 46-year-old woman?

CLUSTER EVALUATION:

- Unsupervised cluster evaluation using cohesion and separation
- Unsupervised cluster evaluation using proximity matrix
- Unsupervised evaluation of hierarchical clustering
- Determining the correct number of clusters
- Clustering Tendency
- Supervised measure of clustering validity
- Assessing the significance of cluster validity measure

K-NN

Classification Approaches:

A) Inductive step: for constructing a classification model from data (decision tree)

B) Deductive step: for applying the model to test examples.

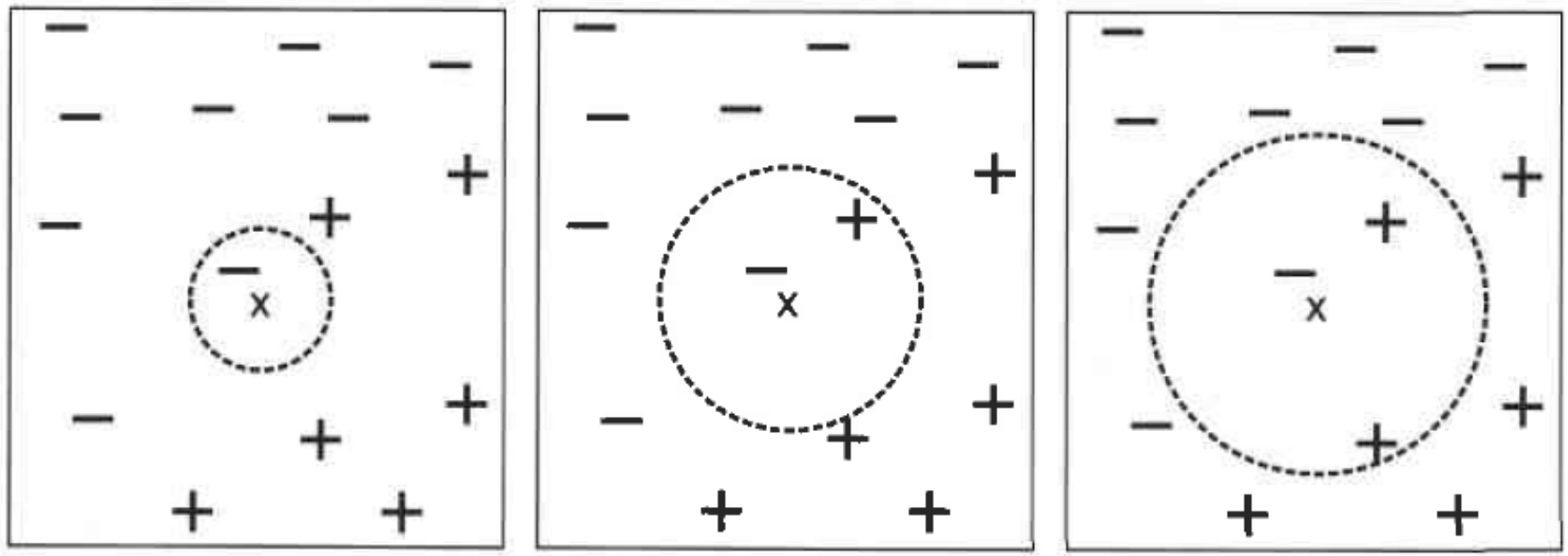
Supervised Machine Learning Algorithm

Solves both regression and classification problems.

Lazy learning algorithm because instead of learning from the training set immediately, the K-NN algorithm stores the dataset and trains from the dataset at the time of classification.

K-NN is also a **non-parametric** algorithm, meaning it does not make any assumptions about the underlying data.

K-NN



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

KNN PSEUDOCODE:

1. Calculate “d(x, xi)” i =1, 2, ….., n; where d denotes the Euclidean distance between the points.
2. Arrange the calculated n Euclidean distances in non-decreasing order.
3. Let k be a +ve integer, take the first k distances from this sorted list.
4. Find those k-points corresponding to these k-distances.
5. Let ki denotes the number of points belonging to the ith class among k points i.e. k ≥ 0
6. If ki >kj ∀ i ≠ j then put x in class i.

K-NN ALGORITHM:

Algorithm 5.2 The k-nearest neighbor classification algorithm.

1: Let k be the number of nearest neighbors and D be the set of training examples.
2: **for** each test example $z = (\mathbf{x}', y')$ **do**
3: Compute $d(\mathbf{x}', \mathbf{x})$, the distance between z and every example, $(\mathbf{x}, y) \in D$.
4: Select $D_z \subseteq D$, the set of k closest training examples to z.
5: $y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)$
6: **end for**

K-NN:

It is ideal for non-linear data and has relatively high accuracy.

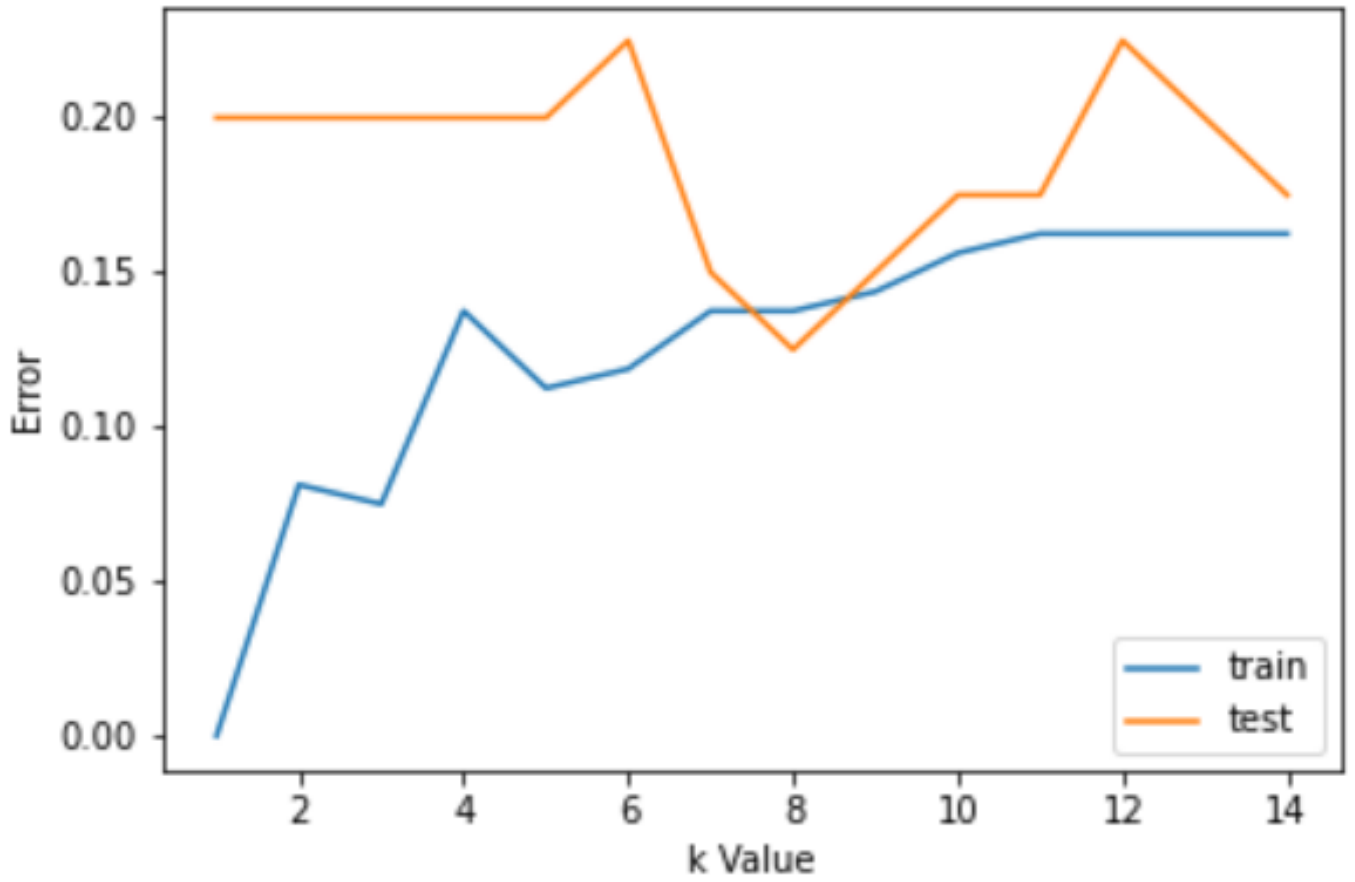
K-NN is the best algorithm if we need to classify data into more than two categories or if the data comprises more than two labels.

Less value of k: susceptible to overfitting

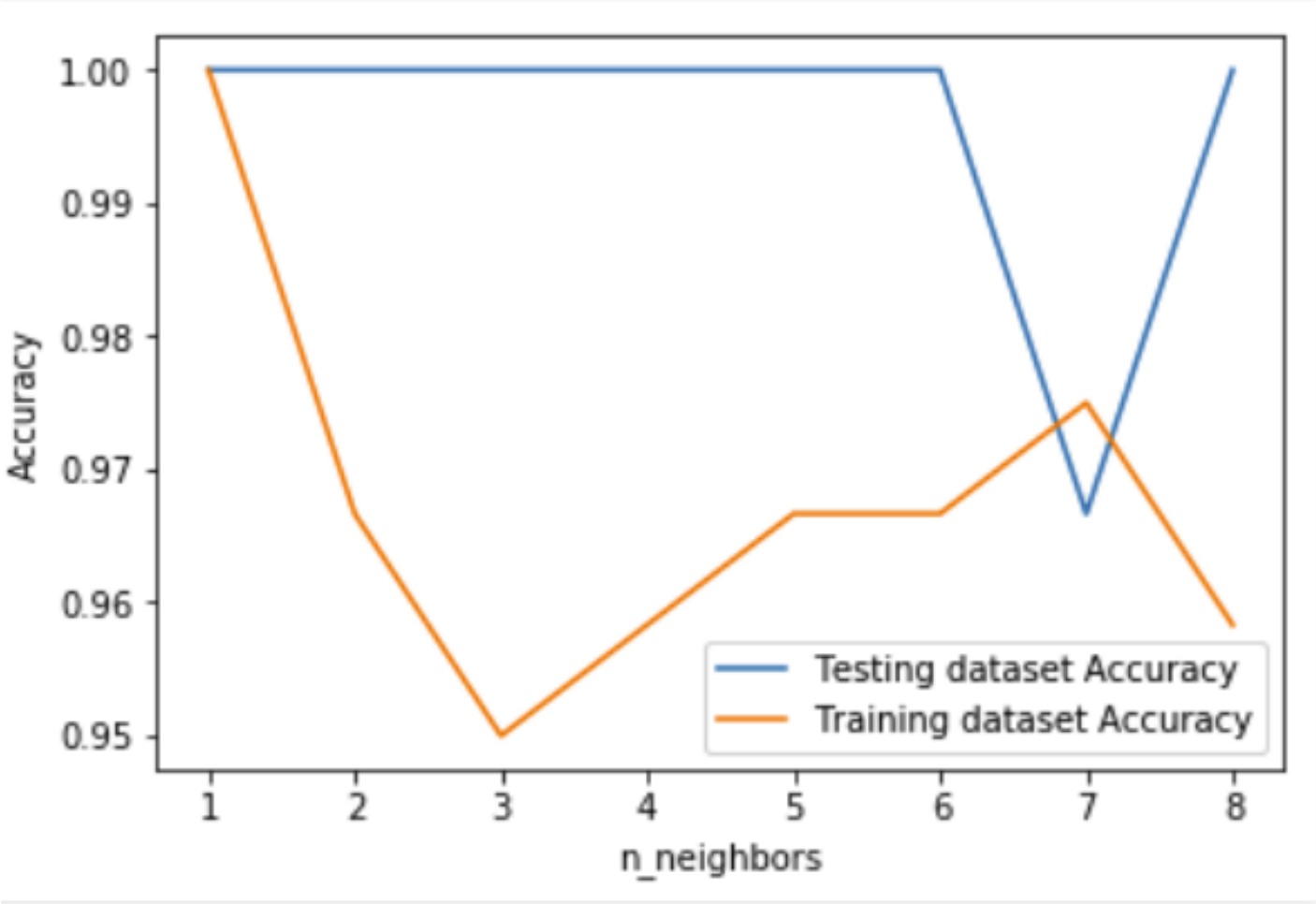
High value of k: a large list of NN may include data points far away from its neighborhood.

$$\text{Distance-Weighted Voting: } y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} w_i \times I(v = y_i).$$

HOW TO SELECT VALUE OF **K:**



HOW TO SELECT VALUE OF **K:**



CHARACTERISTICS OF KNN CLASSIFIER:

- Instance-Based Learning
- Lazy Learner
- Prediction based on local information: susceptible to noise
- Arbitrarily shaped decision boundary: increase in NN may reduce variability
- appropriate proximity measures and pre-processing steps are crucial.

KNN – PROS:

It is very simple algorithm to understand and interpret.

It is very useful for nonlinear data because there is no assumption about data in this algorithm.

It is a versatile algorithm as we can use it for classification as well as regression.

It has relatively high accuracy but there are much better supervised learning models than KNN.

KNN – CONS:

- It is computationally a bit expensive algorithm because it stores all the training data.
- High memory storage required as compared to other supervised learning algorithms.
- Prediction is slow in case of big N.
- It is very sensitive to the scale of data as well as irrelevant features.

KNN – APPLICATION:

- Banking
- Calculating credit rating
- Politics

ANOMALY DETECTION:

- Deviation detection or outliers
- Fraud detection
- Ecosystem disturbances
- Intrusion Detection
- Public health
- Medicine

OUTLIER DETECTION PRELIMINARIES:

- Explore the cause of anomalies
  - Data from different classes
  - Natural deviation
  - Data measurement and collection errors
- Consider various anomaly detection approaches
  - Model Based
  - Proximity Based
  - Density Based

OUTLIER DETECTION PRELIMINARIES:

- Draw distinctions among approaches based on whether they use label information
  - Supervised anomaly detection
  - Unsupervised anomaly detection
  - Semi-supervised anomaly detection
- Common issues with anomaly detection techniques
  - Number of attributes used to define an anomaly
  - Global vs Local perspective
  - Degree to which a point is an anomaly
  - Identifying one anomaly at a point vs many anomalies at once
  - Evaluation and efficiency

OUTLIER DETECTION :

- Probabilistic definition of outliers:
  - *An outlier is an object that has a low probability with respect to a probability distribution model of the data.*
- Issues in outlier detection:
  - Identifying the specific distribution of the data set: heavily tailed data
    - Common distributions: Poisson, gaussian or binomial
    - If the wrong model is chosen then an object can erroneously be identified as an outlier.
  - The number of attributes used:
    - Most of the techniques are for single variables but some are for multivariate data
  - Mixtures of distributions:
    - Distribution needs to be identified before objects can be classified as outliers.
    - Model is based on different data distributions and hence complicated to use and understand.

OUTLIER DETECTION:

- Detecting outliers in univariate normal distribution:
  - Gaussian (normal) distribution: most frequently used in statistics
  - Mean and standard deviation
  - 0.0027 probability that the point is beyond -3 to +3.

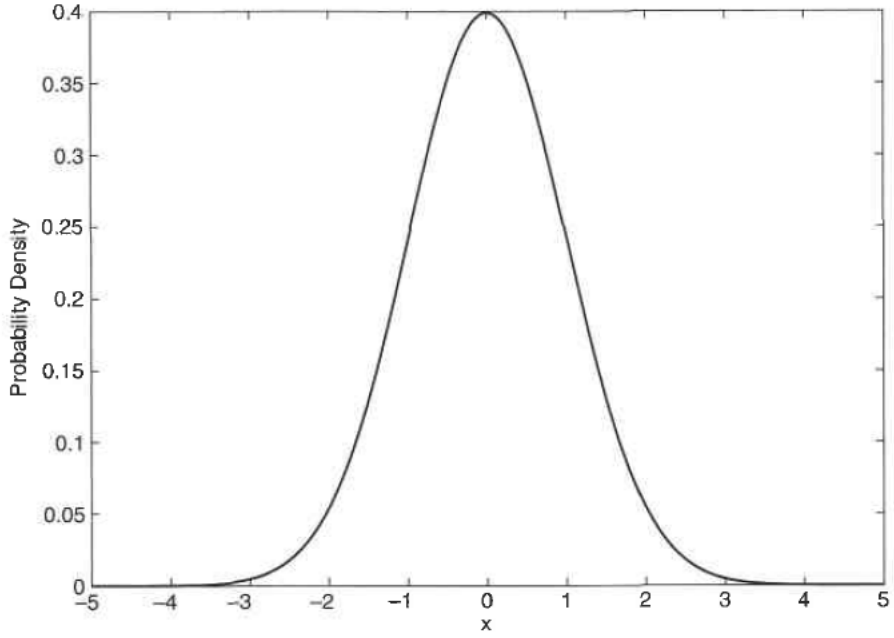


Figure 10.1. Probability density function of a Gaussian distribution with a mean of 0 and a standard deviation of 1.

OUTLIER DETECTION:

- Detecting outliers in multivariate normal distribution:
  - Due to multiple variables it is not symmetrical to the center.
  - Mahalanobis D = log of probability density of the point + constant

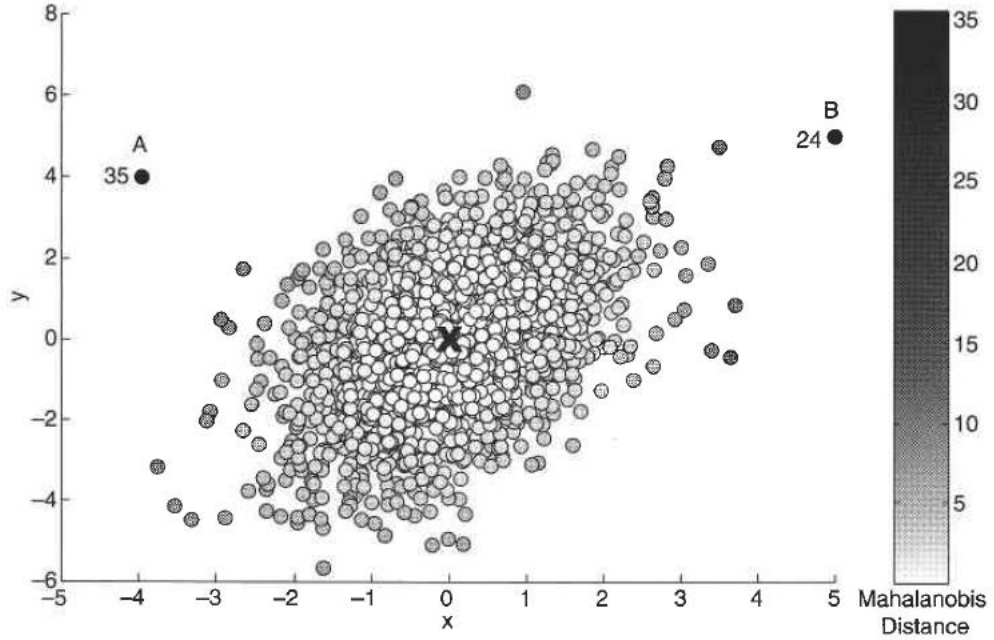


Figure 10.3. Mahalanobis distance of points from the center of a two-dimensional set of 2002 points.

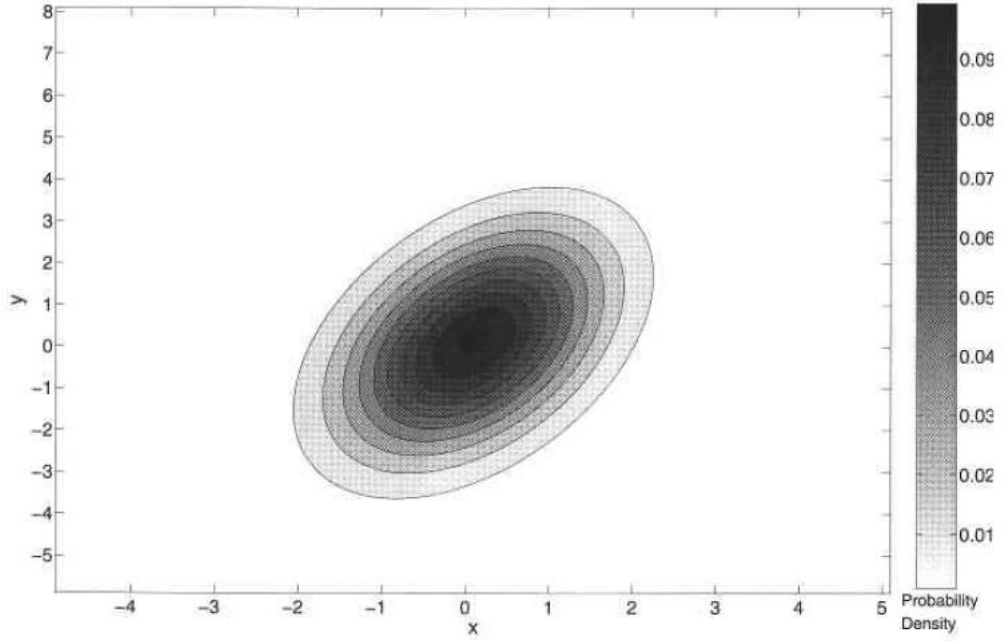


Figure 10.2. Probability density of points for the Gaussian distribution used to generate the points of Figure 10.3.

OUTLIER DETECTION:

- Mixture model approach for anomaly detection:
  - Clustering: data comes from a mixture of probability distributions and each cluster can be identified as one of these distributions.
  - Anomaly: data is modeled as a mixture of two distributions, one for ordinary data and one for outliers.
  - Goal is to maximize the overall likelihood (probability) of data.

Algorithm 10.1 Likelihood-based outlier detection.

1: Initialization: At time $t = 0$, let M_t contain all the objects, while A_t is empty. Let $LL_t(D) = LL(M_t) + LL(A_t)$ be the log likelihood of all the data.
2: **for** each point $\mathbf{x}$ that belongs to M_t **do**
3: Move $\mathbf{x}$ from M_t to A_t to produce the new data sets A_{t+1} and M_{t+1}.
4: Compute the new log likelihood of D, $LL_{t+1}(D) = LL(M_{t+1}) + LL(A_{t+1})$
5: Compute the difference, $\Delta = LL_t(D) - LL_{t+1}(D)$
6: **if** $\Delta > c$, where c is some threshold **then**
7: $\mathbf{x}$ is classified as an anomaly, i.e., M_{t+1} and A_{t+1} are left unchanged and become the current normal and anomaly sets.
8: **end if**
9: **end for**

OUTLIER DETECTION:

- Strength:
  - Statistical approaches to outlier detection have a firm foundation and build on standard statistical techniques, such as estimating the parameters of a distribution.
  - When there is sufficient knowledge of the data and the type of test that should be applied these tests can be very effective. There are a wide variety of statistical outliers tests for single attributes. Fewer options are available for multivariate data.
- Weakness:
  - These tests can perform poorly for high-dimensional data.

PROXIMITY-BASED OD:

- Distance to k-nearest neighbors
- Outlier score can be highly sensitive to the value of k
- Strength
  - Proximity-based approaches typically take O(m2) time. For large data sets, this can be too expensive, although specialized algorithms can be used to improve performance in the case of low dimensional data.
- Weakness
  - The approach is sensitive to the choice of parameters.
  - It cannot handle data sets with regions of widely differing densities because it uses global thresholds that cannot take into account such density variations.

DENSITY-BASED OD:

- closely related to proximity based detection

$$density(\mathbf{x}, k) = \left(\frac{\sum_{\mathbf{y} \in N(\mathbf{x},k)} distance(\mathbf{x}, \mathbf{y})}{|N(\mathbf{x}, k)|} \right)^{-1}$$

CLUSTERING APPROACH FOR OD:

- Clustering: Finds group of strongly related items while anomaly do not
- Discard small clusters that are far away from other clusters
- Assessing the extend to which an object belongs to a cluster
  - Proximity measure
  - Gaussian distribution(Mahalanobis distance)
- Impact of outliers on the initial clustering
- The number of clusters to use
  - Small clusters tends to be more cohesive
  - In large number of small clusters if an object is outlier, then it is true outlier
- Strength and weakness

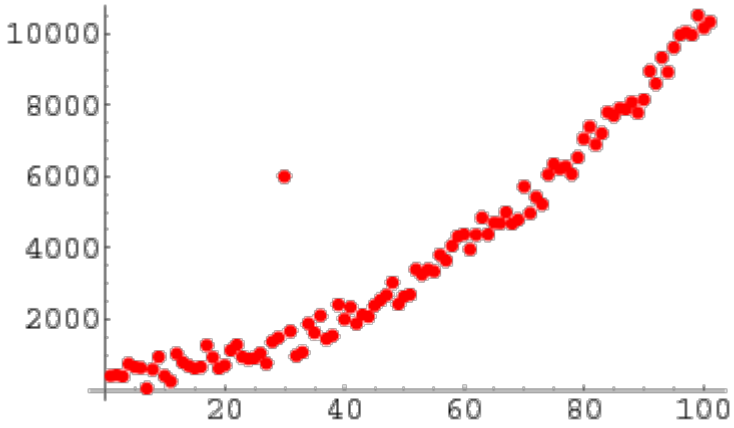
CLUSTERING APPROACH FOR OD:

- Strength
  - Outlier detection technique based on k-means is highly efficient
  - It is usually easy to find both outliers and clusters at the same time.
- Weakness
- The set of outliers produced and their scores can be heavily dependent upon the number of clusters used as well as the presence of outliers in the data

Some Practice Questions

1) **Explain what is Anomaly Detection?**

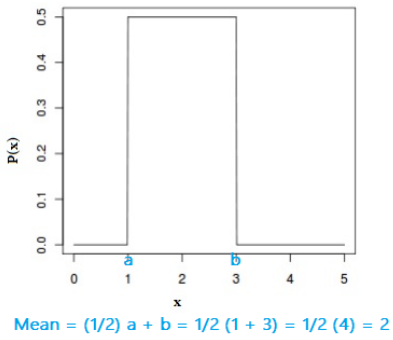
Anomaly detection (or outlier detection) is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data.



2) **What are the differences in *Anomalies* for *Uniform Distribution* and *Normal Distribution* in *One-Dimensional Data*?**

Answer
Uniform Distribution

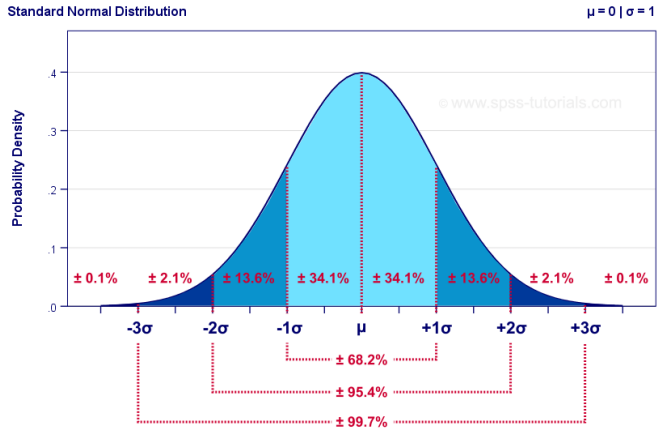
- A *uniform distribution* looks like the following:



- When data is distributed *uniformly* over a finite range, the mean and standard deviation merely characterize the *range of values*.
- One possible indication of *anomalous behavior* could be that a small neighborhood contains *substantially fewer* or *more* data points than expected from a *uniform distribution*.

Normal Distribution

- A normal distribution looks like the following:



- A *normal distribution* follows the *empirical rule*, which states that 68%, 95%, and 99.7% of the values lie within one, two, and three *standard deviations* of the *mean*, respectively.
- About 0.1% of the points are more than 3σ (three standard deviations) away from the mean, and only about 5 x 10^{-8}% of the points are more than six standard deviations away from the mean.
- Hence, a threshold ($3\times\sigma$) is chosen, and points beyond that distance from the mean are declared to be *anomalous*.

3) What are the *Swamping* and *Masking* problems in *Anomaly Detection*?

Answer:

- Since anomalies are rare events, making it very difficult to label them with high accuracy, **swamping** is the phenomenon of **labeling normal events as anomalies**.
- When clustering algorithms are used, the data points belonging to different clusters get merged into one cluster, if the number of segments in the dataset is not known, this causes the outlier cluster to be merged to a cluster with normal data points. This causes the outliers to not be detected. This is defined as **masking**.

4) What's the difference between *Normalisation* and *Standardisation*?

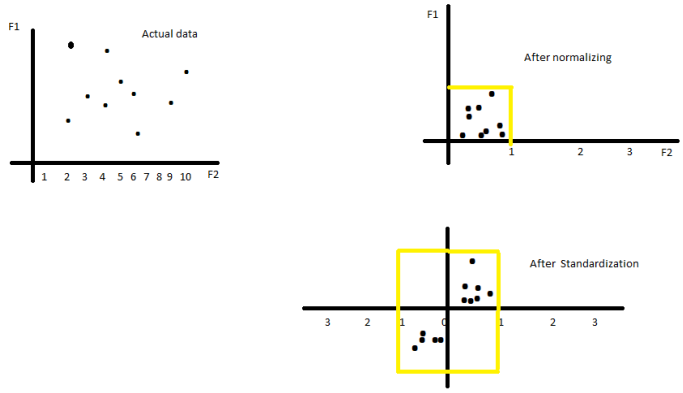
Answer:
Normalization rescales the values into a range of [0,1]. This might be useful in some cases where all parameters need to have the same positive scale. However, the *outliers* from the data set *are lost.*

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Standardization rescales data to have a mean (μ) of 0 and standard deviation (σ) of 1 (unit variance).

$$X_{changed} = \frac{X - \mu}{\sigma}$$

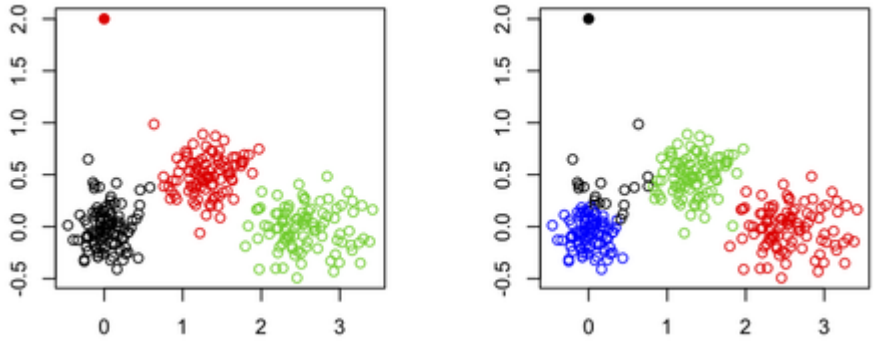
For most applications standardization is recommended.



5) **Can you find *Outliers* using *k-Means*?**

Answer:

- Using **k-Means** for spotting outliers is not good in a multivariate dataset because it is not built for that purpose. It will always give a solution that minimizes the total *within-cluster sum of squares*, and *the outliers will not necessarily define their own cluster*.
- The figure below shows that the *outlying value* is never recovered as such, it always belongs to one of the clusters.



6) **Compare *SVM* and *Logistic Regression* in handling outliers**

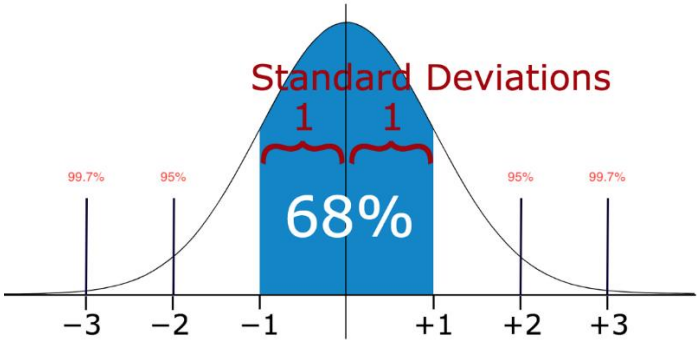
Answer:

- For **Logistic Regression**, outliers can have an *unusually large effect* on the estimate of logistic regression coefficients. It will find a linear boundary if it exists to accommodate the outliers. To solve the problem of outliers, sometimes a sigmoid function is used in logistic regression.

- For **SVM**, outliers can make the decision boundary deviate severely from the optimal hyperplane. One way for SVM to get around the problem is to intrduce *slack variables*. There is a penalty involved with using slack variables, and how SVM handles outliers depends on how this penalty is imposed.

7) **Explain how to use *Standard Deviation* for Anomalies Detection?**

Answer:
In statistics, if a **data distribution** is approximately **normal** then about 68% of the data values lie within **one standard deviation** of the mean and about 95% are within two standard deviations, and **about 99.7%** lie within three standard deviations:



Therefore, if you have any data point that is more than 3 times the standard deviation, then those points are very likely to be **anomalous** or **outliers**.

8) **Explain the difference between *Outlier Detection* vs *Novelty Detection***

Answer

- The training data contains **outliers** which are defined as observations *that are far from the others*. Outlier detection estimators thus try to fit the regions where the training data is the most concentrated, ignoring the deviant observations.
- The training data is not polluted by outliers and we are interested in detecting whether a **new observation** is an outlier. In this context an outlier is also called a **novelty**.

Outlier detection and novelty detection are both used for anomaly detection, where one is interested in detecting abnormal or unusual observations. Outlier detection is then also known as unsupervised anomaly detection and novelty detection as semi-supervised anomaly detection.

In the context of outlier detection, the outliers/anomalies cannot form a dense cluster as available estimators assume that the outliers/anomalies are located in low density regions. On the contrary, in the context of novelty detection, novelties/anomalies can form a dense cluster as long as they are in a low density region of the training data, considered as normal in this context.

9) **What are some *categories* of outlier detection approaches?**

Answer
There are three categories of outlier detection, namely, *supervised*, *semi-supervised*, and *unsupervised*:

- **Supervised**: Requires fully labeled training and testing datasets. An ordinary classifier is trained first and applied afterward.
- **Semi-supervised**: Uses training and test datasets, whereas training data only consists of normal data without any outliers. A model of the normal class is learned and outliers can be detected afterward by deviating from that model.
- **Unsupervised**: Does not require any labels; there is no distinction between a training and a test dataset Data is scored solely based on intrinsic properties of the dataset.

And three fundamental approaches to detect anomalies are based on:

- **By Density** - Normal points occur in dense regions, while anomalies occur in sparse regions
- **By Distance** - Normal point is close to its neighbors and anomaly is far from its neighbors
- **By Isolation** - The term isolation means ‘separating an instance from the rest of the instances’. Since anomalies are ‘few and different’ and therefore they are more susceptible to isolation.

10) **What is the difference between *Out of Distribution* and *Anomaly Detection*?**

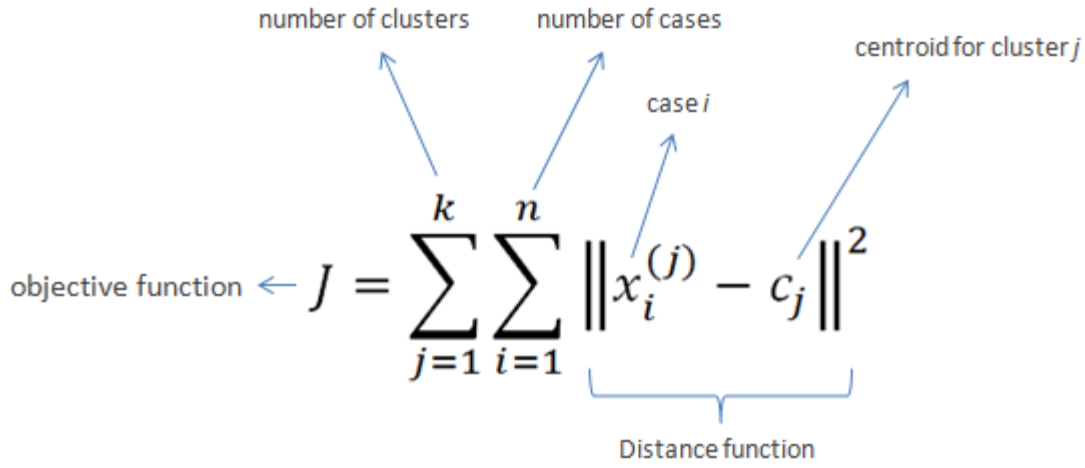
Answer

- **Out of distribution** (*OOD*) data refers to data that was collected at a different time, and possibly under different conditions or in a different environment than the data collected to create the model. It can be said that the data is from a *different distribution*.
- After the *out of distribution data* is collected, the model can perform either **Novelty detection** or **Anomaly detection**.
- **Novelty** data is the data that is in-distribution. Novelty detection checks whether the new data is in-distribution or not.
- **Anomaly detection** is used to test the data to see if it is different than what the model was trained on.

11) Explain the steps of k-Means Clustering Algorithm

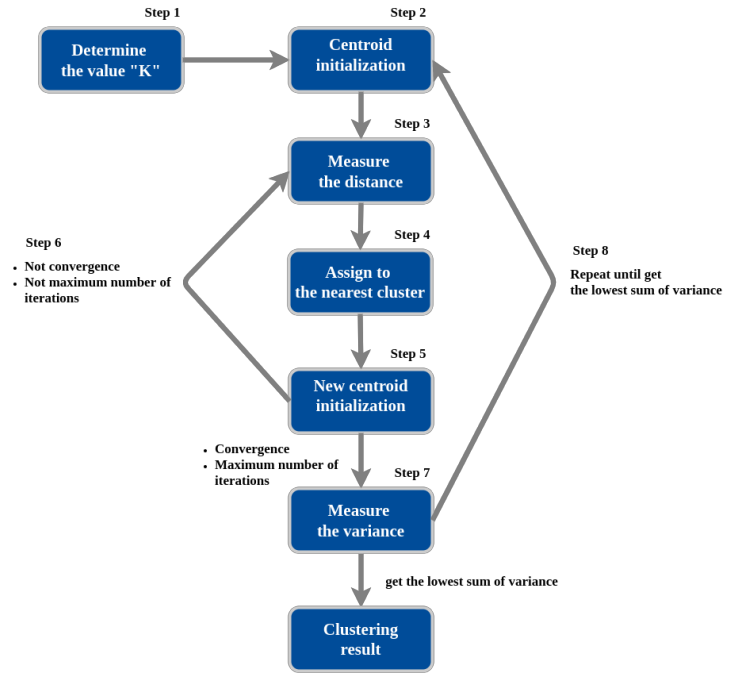
Answer

K-Means clustering intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly *k* different clusters of the greatest possible distinction. The best number of clusters *k* leading to the greatest separation (distance) is not known as a priori and must be computed from the data. The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function:



Algorithm:

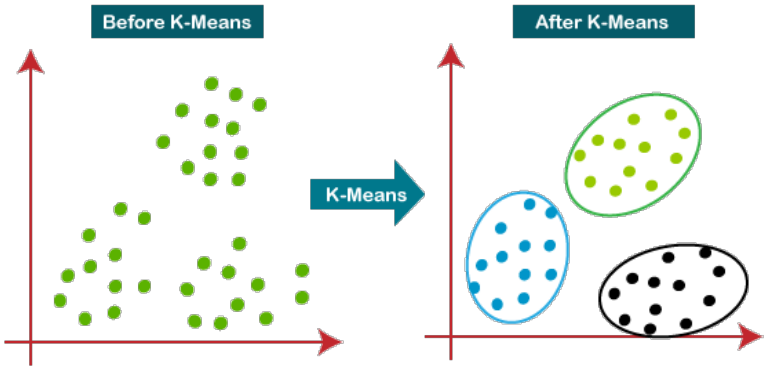
1. Clusters the data into *k* groups where *k* is predefined.
2. Select *k* points at *random* as cluster centers.
3. Assign objects to their closest cluster center according to the *Euclidean distance* function.
4. Calculate the *centroid* or *mean* of all objects in each cluster.
5. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.



12) Explain what is *k-Means Clustering*?

Answer

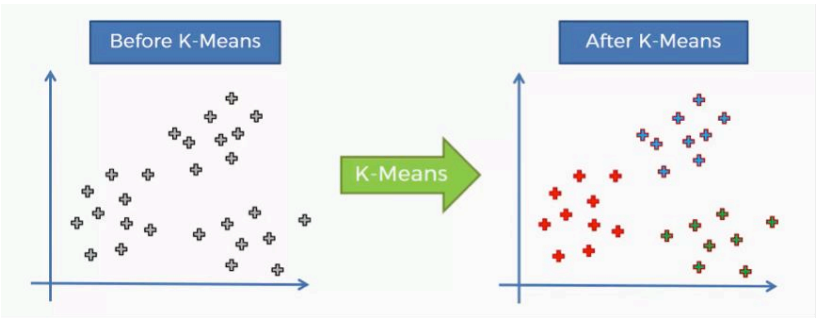
- **k-means clustering** is a method of *vector quantization* that aims to partition n observations into k clusters in which each observation belongs to the *cluster* with the nearest *mean*.
- k-means clustering minimizes **within-cluster variances**.
- **Within-cluster-variance** is simple to understand **measure of compactness**. So basically, the objective is to find the most *compact* partitioning of the data set into k partitions.



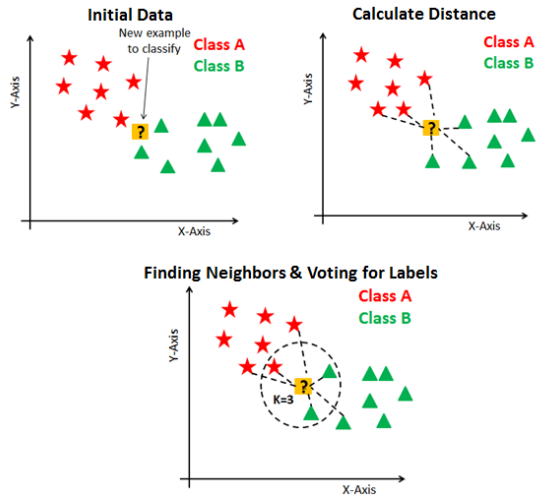
13) What is the main difference between *k-Means* and *k-Nearest Neighbours*?

Answer

- **k-Means** is a *clustering algorithm* that tries to partition a set of points into k sets such that the points in each cluster tend to be near each other. It is *unsupervised* because the points have no external classification.



- **k-Nearest Neighbors** is a *classification* (or *regression*) algorithm that, in order to determine the classification of a point, combines the classification of the k nearest points. It is *supervised* because it is trying to classify a point based on the known classification of other points.



14) Compare *Hierarchical Clustering* and *k-Means Clustering*

Answer:
Scalability

- **k-means** has an order of $O(n.k.d.i)$. The scalability of k-means is better than hierarchical clustering because k, i, and d are small, and also the memory consumption is linear.
- **Hierarchical clustering** has an order of $O(n^3d)$. The scalability of hierarchical clustering is worse, and its memory consumption is quadratic.

Flexibility

- **k-means** is extremely limited in applicability. It is limited to *Euclidean distances*, and it only works on *numerical data*.
- **Hierarchical clustering** does not even require distances (any measure can be used, including similarity function simply by preferring high values to low values). It can use any type of data including categorical, strings, time series, or mixed.

15) Explain some cases where *k-Means clustering* fails to give good results

Answer

- ***k-means*** has trouble clustering data where clusters are of *various sizes* and *densities*.
- **Outliers** will cause the *centroids to be dragged*, or the outliers might get their own cluster instead of being ignored. Outliers should be clipped or removed before clustering.
- If the number of dimensions increase, a distance-based similarity measure converges to a constant value between any given examples. *Dimensions should be reduced* before clustering them.

16) How is *Entropy* used as a *Clustering Validation Measure*?

Answer

- ***Entropy*** is used as an *external validation* measure by using the class labels of data as external information.

- **Entropy** is a measure of the purity of the cluster with respect to the given class label. Thus, if each cluster consists of objects with a single class label, the entropy value is 0. As the objects in a cluster become more diverse, the entropy value increases.
- The *entropy* of a cluster j is calculated by:

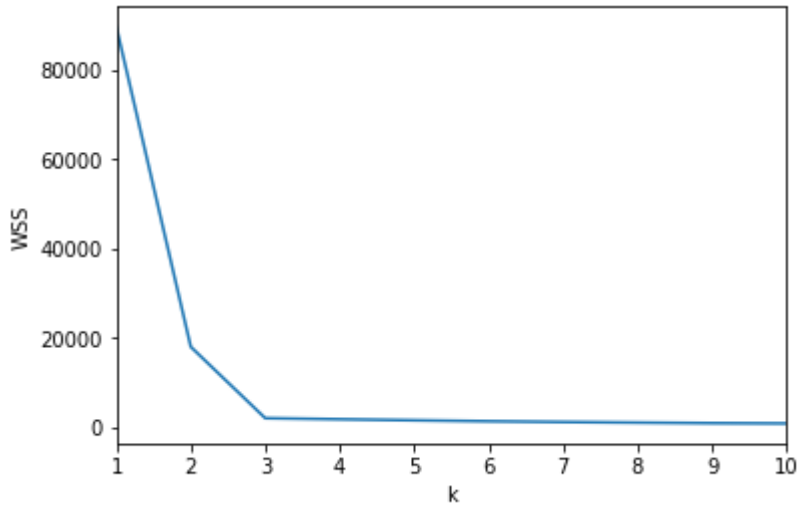
$$E_j = -\sum_i p_{ij} \log(p_{ij})$$

where *pij* is the probability of assigning an object of class i to cluster j, and the sum is taken over all classes.

- Using *entropy measure* to validate the class labels tends to favor *k-means* which produce clusters in relatively uniform size. This effect is more significant in the situation that the data have highly imbalanced true clusters.
- So, using *entropy measure* for validating *k-means* clustering can lead to the results being misleading.

17) How to determine k using the Elbow Method?

Answer
Calculate the **Within-Cluster-Sum of Squared** Errors (WSS) for **different values of k**, and choose the k for which WSS becomes first starts to diminish. In the plot of WSS-versus-k, this is visible as an **elbow.**



18) How would you *Pre-Process* the data for *k-Means*?

Answer
Some *pre-processing* steps to follow are:

- If the variables are of *incomparable units*, then the variables should be **standardized**.
- Even if the variables are of the same units but show quite different *variances* then it is a good idea to **standardize** them. Since *k-means* clustering produces more or less round clusters, it puts more weight on variables with smaller variance, so the clusters will tend to be separated along with variables with greater variance.
- *k-means* clustering results are sensitive to the *order of objects* in the dataset, so it is good to **randomize the dataset** and try clustering many different times.

19) How would you perform *k-Means* on very large datasets?

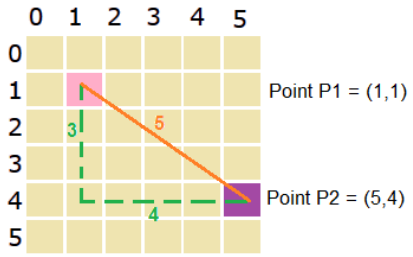
Answer

- If the dataset is large, an alternative is using **mini-batch k-Means**.
- **Mini batch k-means** has the main advantage of reducing the computational cost of finding a partition. This cost is proportional to the size of the sample batch used and this difference is more evident when the *number of clusters is larger*.
- The main idea of **mini-batch k-Means** is to use small random batches of data of a fixed size, so they can be stored in memory. Each iteration a new random sample from the dataset is obtained and used to update the clusters and this is repeated until convergence.

20) What is the difference between the *Manhattan Distance* and *Euclidean Distance* in Clustering?

Answer

- **Manhattan distance** captures the distance between two points by *aggregating the pairwise absolute difference* between each variable.
- **Euclidean distance** captures the distance between two points by *aggregating the squared difference* in each variable.
- If two points are close on most variables but more discrepant on one of them, **Euclidean distance** will exaggerate that discrepancy, whereas **Manhattan distance** will shrug it off, being more influenced by the closeness of the other variables.
- **Manhattan distance** should give more robust results, whereas **Euclidean distance** is likely to be influenced by outliers.



Euclidean distance = $\sqrt{(5-1)^2 + (4-1)^2} = 5$

Manhattan distance = $|5-1| + |4-1| = 7$

21) While performing *K-Means* Clustering, how do you determine the value of *K*?

Answer
There are many different approaches to find the value of K. Some of the approaches are described below:

- Maximizing the **Bayesian Information Criterion** (BIC):

$$BIC(C|X) = L(X|C) - (p/2).\log n$$

where L(X|C) is the *log-likelihood* of the dataset X according to model C. p is the number of parameters in the model C, and n is the number of points in the dataset.

- Another method is to start with a large value of k and removing *centroids* (reducing k) until it no longer reduces the description length.
- Another method is to start with *one* cluster, and split the clusters until the points assigned to each cluster has a *Gaussian* distribution.

PRACTICE PROBLEMS BASED ON K-MEANS CLUSTERING ALGORITHM-

Problem: Cluster the following eight points (with (x, y) representing locations) into three clusters:
A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2). The distance function between two points a = (x1, y1) and b = (x2, y2) is defined as- P(a, b) = |x2 – x1| + |y2 – y1|. Use K-Means Algorithm to find the three cluster centers after the second iteration.

Solution- We follow the above discussed K-Means Clustering Algorithm-

Iteration-01:

- We calculate the distance of each point from each of the center of the three clusters.
- The distance is calculated by using the given distance function.

The following illustration shows the calculation of distance between point A1(2, 10) and each of the center of the three clusters-

Calculating Distance Between A1(2, 10) and C1(2, 10)-

P(A1, C1)

= |x2 – x1| + |y2 – y1|

= |2 – 2| + |10 – 10|

= 0

Calculating Distance Between A1(2, 10) and C2(5, 8)-

P(A1, C2)

= |x2 – x1| + |y2 – y1|

= |5 – 2| + |8 – 10|

= 3 + 2

= 5

Calculating Distance Between A1(2, 10) and C3(1, 2)-

P(A1, C3)

= |x2 – x1| + |y2 – y1|

= |1 – 2| + |2 – 10|

= 1 + 8

= 9

In the similar manner, we calculate the distance of other points from each of the center of the three clusters.

Next,

- We draw a table showing all the results.
- Using the table, we decide which point belongs to which cluster.
- The given point belongs to that cluster whose center is nearest to it.

Given Points	Distance from the center (2, 10) of Cluster-01	Distance from the center (5, 8) of Cluster-02	Distance from the center (1, 2) of Cluster-03	Point belongs to Cluster

A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

From here, New clusters are-

Cluster-01: First cluster contains points-

- A1(2, 10)

Cluster-02: Second cluster contains points-

- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)
- A8(4, 9)

Cluster-03: Third cluster contains points-

- A2(2, 5)
- A7(1, 2)

Now,

- We re-compute the new cluster clusters.
- The new cluster center is computed by taking mean of all the points contained in that cluster.

For Cluster-01:

- We have only one point A1(2, 10) in Cluster-01.
- So, cluster center remains the same.

For Cluster-02:

Center of Cluster-02

= ((8 + 5 + 7 + 6 + 4)/5, (4 + 8 + 5 + 4 + 9)/5)

= (6, 6)

For Cluster-03:

Center of Cluster-03

= ((2 + 1)/2, (5 + 2)/2)

= (1.5, 3.5)

This is completion of Iteration-01.

Iteration-02:

- We calculate the distance of each point from each of the center of the three clusters.
- The distance is calculated by using the given distance function.

The following illustration shows the calculation of distance between point A1(2, 10) and each of the center of the three clusters-

Calculating Distance Between A1(2, 10) and C1(2, 10)-

P(A1, C1)

$= |x2 – x1| + |y2 – y1|$

$= |2 – 2| + |10 – 10|$

$= 0$

Calculating Distance Between A1(2, 10) and C2(6, 6)-

P(A1, C2)

$= |x2 – x1| + |y2 – y1|$

$= |6 – 2| + |6 – 10|$

$= 4 + 4$

$= 8$

Calculating Distance Between A1(2, 10) and C3(1.5, 3.5)-

P(A1, C3)

$= |x2 – x1| + |y2 – y1|$

$= |1.5 – 2| + |3.5 – 10|$

$= 0.5 + 6.5$

$= 7$

In the similar manner, we calculate the distance of other points from each of the center of the three clusters.

Next,

- We draw a table showing all the results.
- Using the table, we decide which point belongs to which cluster.
- The given point belongs to that cluster whose center is nearest to it.

Given Points	**Distance from center (2, 10) of Cluster-01**	**Distance from center (6, 6) of Cluster-02**	**Distance from center (1.5, 3.5) of Cluster-03**	**Point belongs to Cluster**
A1(2, 10)	0	8	7	C1
A2(2, 5)	5	5	2	C3
A3(8, 4)	12	4	7	C2

A4(5, 8)	5	3	8	C2
A5(7, 5)	10	2	7	C2
A6(6, 4)	10	2	5	C2
A7(1, 2)	9	9	2	C3
A8(4, 9)	3	5	8	C1

From here, New clusters are-

Cluster-01: First cluster contains points-

- A1(2, 10)
- A8(4, 9)

Cluster-02: Second cluster contains points-

- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)

Cluster-03: Third cluster contains points-

- A2(2, 5)
- A7(1, 2)

Now,

- We re-compute the new cluster clusters.
- The new cluster center is computed by taking mean of all the points contained in that cluster.

For Cluster-01: Center of Cluster-01
= ((2 + 4)/2, (10 + 9)/2)

= (3, 9.5)

For Cluster-02: Center of Cluster-02
= ((8 + 5 + 7 + 6)/4, (4 + 8 + 5 + 4)/4)

= (6.5, 5.25)

For Cluster-03: Center of Cluster-03
= ((2 + 1)/2, (5 + 2)/2)

= (1.5, 3.5)

This is completion of Iteration-02.

After second iteration, the center of the three clusters are-

- C1(3, 9.5)
- C2(6.5, 5.25)
- C3(1.5, 3.5)