OPTIMIZATION TECHNIQUES AND DECISION MAKING	
Course Code: BAM-301 Contact Hours: L-3 T-0 P-2 Course Category: DCC	Credits: 4 Semester: 5

Introduction: Optimization Techniques are specific method to achieve the minima and maxima or the optimizing problems. Decisions making in businesses, research and scientific domains often strive to solve the optimizing problems for achieving the underlying objectives of the businesses, research and scientific domains. This course introduces several optimization techniques and their applications in computer science domain.

Course Objectives:

- To study Linear Programming and Integer Programming Problems
- To learn constraints satisfaction and its application aspects in engineering problems
- To apply data analysis and harness in decision makings
- To be able to apply optimal solutions in variety of engineering domains

Pre-requisite: Basic Mathematics

Course Outcomes: After completion of the course, students will be able to:

CO1: To be able to understand LP paradigm and pure & mixed Integer Problems
CO2: Understand the basic concepts of the Constraint Satisfaction and Applications
CO3: Understand Classification, Clustering and Regression Queuing Theory
CO4: To be able to understand Analysis and Decision making aspects

Pedagogy: Classroom teaching which focuses on developing understanding of students to digest the concepts of subject with large number of examples. The teaching-learning of the course would be organized through lectures, tutorials, assignments, and quizzes.

Contents

UNIT-I	10 Hours
Fundamental theorem of linear programming, Degenerate solutions, Simplex based methods, Cycling, Duality, Complementary slackness conditions. Non-linear programming: First and second order conditions. Iterative methods and associated issues, Line search methods: Stationarity of limit points of steepest decent, successive step-size reduction algorithms, etc.	
UNIT-II	**10 Hours**
Hessian based algorithms: Newton, Conjugate directions and Quasi-Newton methods. Constrained optimization problems: Lagrange variables, Karush-Kuhn-Tucker conditions, Regular points, Sensitivity analysis. Quadratic programming, Convex problem	
UNIT-III	**12 Hours**
Prescriptive Analytics, decision support systems, Statistical learning techniques, including regression, Logistic Regression, Ridge Regression, Lasso Regression, K Nearest Neighbors Regression & Classification Methods, Bias-Variance Dichotomy Model Validation Approaches	
UNIT-IV	**10 Hours**
Linear Discriminant Analysis, Quadratic Discriminant Analysis, Regression and Classification Trees, Support Vector Machines, Clustering, Analytic Hierarchy Process (AHP), Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), Reinforcement learning: Introduction, Applications, Markov decision process, Q-learning, SARSA, DQN. Optional Topics: Mixed integer models; Interior point methods; Iterative schemes for constrained problems; Sequential quadratic programming methods; Barrier methods; Trust-region methods	

Text Books	
1	Jiawei Han, Micheline Kamber, Jian Pei. Data Mining: Concepts and Techniques, 3rd Edition, MK publisher, 2011.
2	Taha, H.A. Operations Research, 5th ed., Macmillan Publishing Company, 1992.
3	Mustafi, C. K. Operations Research, 4th ed., New Age International, 2009
Reference Books	
1	Smith, David K. Network Optimization in Practice. Ellis Harwood Publications, 1982.
2	Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. Introduction to Data Mining, 2nd ed., Pearson Education, 2021.

UNIT- 1 Topics for the Mid Sem exam - Fundamental theorem of linear programming, Degenerate solutions, Simplex based methods, Cycling, Duality. Non-linear programming: First and second order conditions. Iterative methods and associated issues (of Simplex Method).

UNIT- 2 Topics for the Mid Sem exam - Constrained optimization concepts, Regular points, Sensitivity analysis. Quadratic programming, Convex problem (Constraint Surface explanation).

BAM 202 MACHINE LEARNING SYLLABUS OF BTECH AI-ML

CONTENTS

UNIT I	10 hours
Introduction: Goals and applications of machine learning. Types of Machine Learning: Supervised Learning, Unsupervised Learning, Machine Learning Cycle: Train-Test Split, Validation Data, K-Fold Cross Validation, Evaluation Metrics. Data Exploration and Pre-processing: Data Objects and Attributes; Statistical Measures, Visualization, Data Cleaning and Integration. Feature Extraction and Reduction.	
UNIT II	**10 hours**
Supervised Learning Regression: Least Mean Square Regression; Ridge Regression and LASSO regression; Logistic Regression, Support Vector Machines. Kernels for learning non- linear functions, K-nearest-neighbor, Bayesian and Naïve Bayes Classifier. Decision Tree Learning.	
UNIT III	**10 hours**
Unsupervised Learning Learning from unclassified data. Clustering. Hierarchical Agglomerative Clustering, k-means partitional clustering, Hierarchical, and Density-based Clustering, Expectation maximization (EM) for soft clustering. Dimensionality Reduction: Linear Discriminant Analysis; Principal Component Analysis;	
UNIT IV	**10 hours**
Advanced Topics Measuring the accuracy of learned hypotheses. Comparing learning algorithms: cross-validation, learning curves, and statistical hypothesis testing, Ensemble Learning: Bagging, boosting, and stacking, Random Forests, Ensemble Classification including Adaboost, Active learning with ensembles.	
Text Books	
1	Han, J., Pei, J. and Tong, H., 2022. Data mining: concepts and techniques. Morgan kaufmann
2	Daumé, H. III, "A Course in Machine Learning", 2015 (freely available online).
3	Mitchell, T. "Machine Learning", 1997 (freely available online)
Reference Books	
1	Shai Shalev-Shwartz and Shai Ben-David. "Understanding Machine Learning: From Theory to Algorithms", Cambridge University Press, 2014
2	Marsland, S., 2011. Machine learning: an algorithmic perspective. Chapman and Hall/CRC.

PROBLEM 1:

A company owns two flour mills viz. A and B, which have different production capacities for high, medium and low quality flour.

The company has entered a contract to supply flour to a firm every month with at least 8, 12 and 24 quintals of high, medium and low quality respectively.

It costs the company Rs.2000 and Rs.1500 per day to run mill A and B respectively.

On a day, Mill A produces 6, 2 and 4 quintals of high, medium and low quality flour, Mill B produces 2, 4 and 12 quintals of high, medium and low quality flour respectively.

How many days per month should each mill be operated in order to meet the

contract order most economically?

Solution:

Let us define x_1 and x_2 as the quantity produced by mills A and B resp. Here the objective is to minimize the cost of the machine runs and to satisfy the contract order.

The LPP problem is given by:

Minimize $Z = 2000x_1 + 1500x_2$

Subject to:

$6x_1 + 2x_2 \geq 8$

$2x_1 + 4x_2 \geq 12$

$4x_1 + 12x_2 \geq 24$

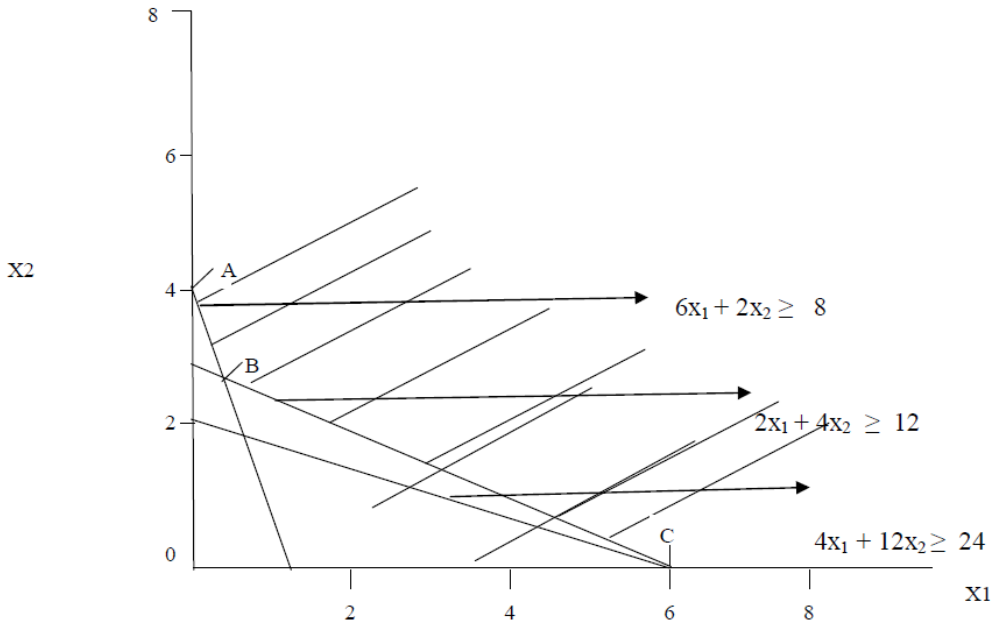
$x_1 \geq 0, x_2 \geq 0$

The three lines $6x_1 + 2x_2 = 8$, $2x_1 + 4x_2 = 12$, and $4x_1 + 12x_2 = 24$ passes through the point

(1.33,0) (0,4), (6,0) (0,3) and (6,0) (0,2) respectively.

The feasible region is shown in the following Graph 2.

In this problem the constraints are of ≥ type of feasible region, which is bounded on one side only.



Graph 2: Feasible Region

Graphical Linear Programming Solution

A two variable linear programming problem can be easily solved graphically. The

method is simple but the principle of solution depends on certain analytical concepts, they are:

Convex Region:

A region R is convex if and only if for any two points on the region R the line connecting those points lies entirely in the region R.

Extreme Point:

The extreme point E of a convex region R is a point such that it is not possible to locate two distinct points in R, so that the line joining them will include E. The extreme points are also called as corner points or vertices.

Thus, the following result provides the solution to the linear programming model:

"If the minimum or maximum value of a linear function defined over a convex region exists, then it must be on one of the extreme or corner points".

The feasible region for this problem is illustrated in Graph 2. Here each of the half planes lies above its boundary. In this case the feasible region is infinite. In this case, we are concerned with the minimization; also it is not possible to determine the maximum value.

Extreme Point	Coordinates		Objective Function
	X_1	X_2	$2000x_1 + 1500x_2$
A	$X_1 = 0$	$X_2 = 4$	6000
B	$X_1 = 0.5$	$X_2 = 2.75$	5125
C	$X_1 = 6$	$X_2 = 0$	12000

Table 2: Shows the objective function Minimum value computation

The minimum value is 5125 at the extreme point B, which is the value of the M (objective function). The optimum values variables are $X_1 = 0.5$ and $X_2 = 2.75$.

DECISION TREE ANALYSIS

Decision tree is the graphical display of the progression of decision and random events.

A decision tree analysis involves the construction of a diagram that shows, at a glance, when decisions are expected to be made – in what sequence, their possible outcomes, and the corresponding payoffs.

A decision tree consists of nodes, branches, probability estimates, and payoffs. The decision tree utilizes probability factors as a means of arriving at a final answer.

There are two types of nodes:

a) Decision (or act) node: A decision node is represented by a square and represents a point of time where a decision-maker must select one alternative course of action among the available. The courses of action are shown as branches or arcs emerging out of decision node.

b) Chance (or event / outcome) node: Each course of action may result in a chance node, which corresponds tio States of nature.. The chance node is represented by a circle and indicates a point of time where the decision-maker will discover the response to his decision.

Branches emerge from and connect various nodes and represent either decisions or states of nature. There are two types of branches:

i. Decision branch: It is the branch leading away from a decision node and represents a course of action that can be chosen at a decision point.

ii. Chance branch: It is the branch leading away from a chance node and represents the state of nature of a set of chance events. The assumed probabilities of the states of nature are written alongside their respective chance branch.

iii. Terminal branch: Any branch that makes the end of the decision tree (not followed by either a decision or chance node), is called a terminal branch. A terminal branch can represent either a course of action. The terminal points of a decision tree are supposed to be mutually exclusive points so that exactly one course of action will be chosen

The payoff can be positive (i.e. revenue or sales) or negative (i.e. expenditure or cost) and it can be associated either with decision or chance branches.

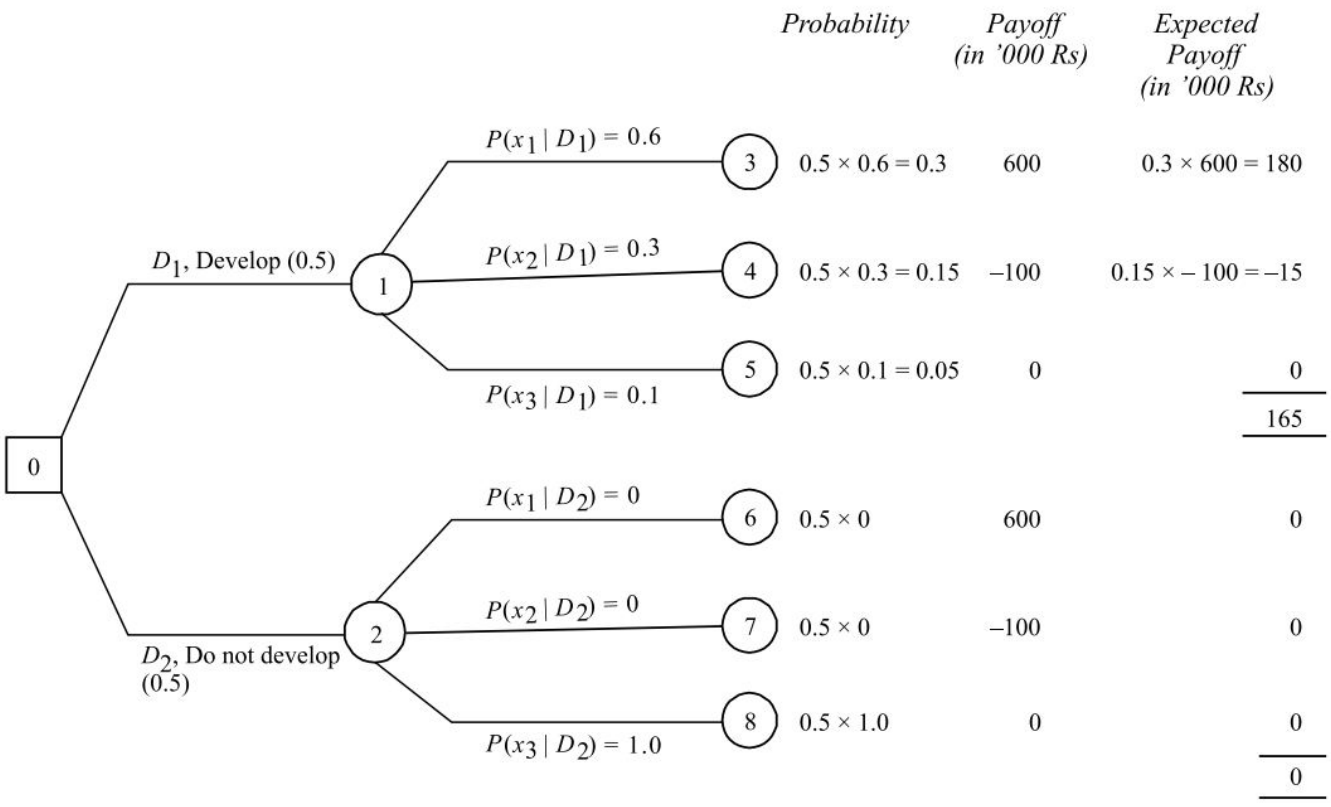
The most important feature of the decision tree, is that it takes time differences of future earnings into account. At any stage of the decision tree, it may be necessary to weigh differences in immediate cost or revenue against differences in value at the next stage.

Example 1-Given the following estimates concerning a R&D programme:

Decision D_i	*Probability of Decision* D_i *Given Research R* $P(D_i \mid R)$	*Outcome Number*	*Probability of Outcome* x_i *Given* D_i $P(x_i \mid D_i)$	*Payoff Value of Outcome,* x_i *(Rs '000)*
Develop	0.5	1	0.6	600
		2	0.3	– 100
		3	0.1	0
Do not develop	0.5	1	0.0	600
		2	0.0	– 100
		3	1.0	0

Construct and evaluate the decision tree diagram for the above data.

Show your workings for evaluation.

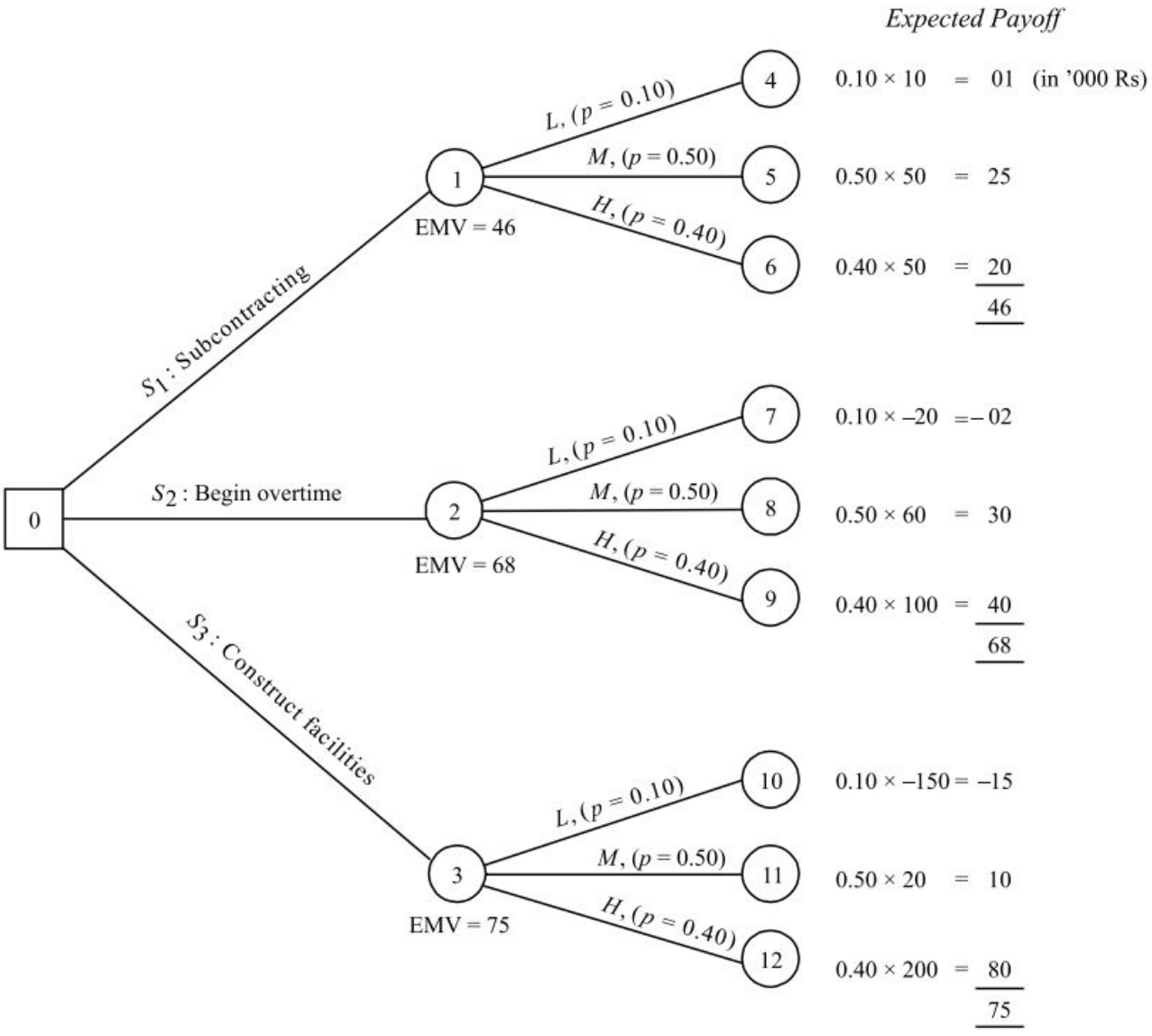


Example 2 : A glass factory that specializes in crystal is developing a substantial backlog and for this the firm's management is considering three courses of action: To arrange for subcontracting (S_1), to begin overtime production (S_2), and to construct new facilities (S_3). The correct choice depends largely upon the future demand, which may be low, medium, or high. By consensus, management ranks the respective probabilities as 0.10, 0.50 and 0.40. A cost analysis reveals the effect upon the profits. This is shown in the table below:

Demand	*Probability*	*Course of Action*		
		S_1 (*Subcontracting*)	S_2 (*Begin Overtime*)	S_3 (*Construct Facilities*)
Low (L)	0.10	10	– 20	– 150
Medium (M)	0.50	50	60	20
High (H)	0.40	50	100	200

Show this situation in the form of a Decision Tree and indicate the optimal decision and its corresponding expected value.

SOLUTION: Calculate expected value of each branch and select the path (course of action) that has highest value.

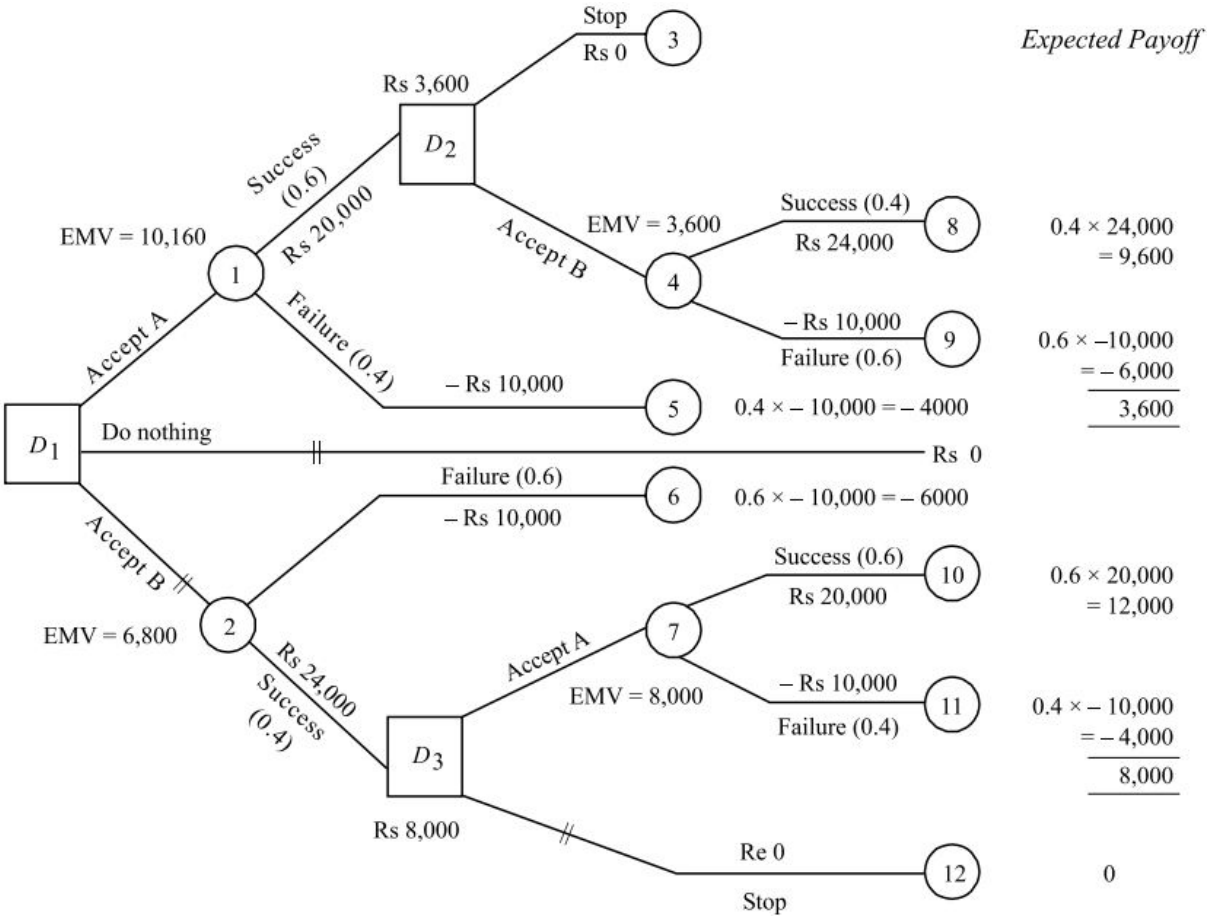


Example 3-A businessman has two portfolios A and B, available to him, but he lacks the capital to undertake both of them simultaneously. He can either choose A first and then stop, or if A is not successful, then take, B or vice versa. The probability of success of A is 0.6, while for B it is 0.4.

Both schemes require an initial budget of Rs 10,000 and both return nothing if venture is unsuccessful. Successful completion of A will return Rs 20,000 (over cost) and successful completion of B will return Rs 24,000 (over cost). Draw a decision tree in order to determine the best strategy.

Solution: The evaluation of each chance node and decision is given in table below

	Decision Point	*Outcome*	*Probability*	*Conditional Value (Rs)*	*Expected Value*
D_3	(i) Accept *A*	Success	0.6	20,000	12,000
		Failure	0.4	–10,000	– 4,000
					8,000
	(ii) Stop	–	–	–	0
D_2	(i) Accept *B*	Success	0.4	24,000	9,600
		Failure	0.6	–10,000	– 6,000
					3,600
	(ii) Stop	–	–	–	0
D_1	(i) Accept *A*	Success	0.6	20,000 + 3,600 = 23,600	14,160
		Failure	0.4	–10,000	– 4,000
					10,160
	(ii) Accept *B*	Success	0.4	24,000 + 8,000 = 32,000	12,800
		Failure	0.6	–10,000	– 6,000
					6,800
	(iii) Do nothing	–	–	–	0



Since the EMV = Rs 10,160 at node D1 is highest, therefore the best strategy is to accept course of action A first and if A is successful, then accept B.

Concept of Regression Tree: Regression Tree is a type of Decision Tree used when the target (output) variable is continuous (not categorical).

It divides the data into smaller and smaller regions so that the value of the dependent variable (Y) within each region is as homogeneous as possible.

How It Works

1. Start with all the training data.
2. At each node, select the variable and the split point that minimizes the sum of squared errors (SSE) or variance within the resulting groups.
3. Repeat splitting recursively until a stopping condition is met (e.g., minimum number of samples, or no significant improvement).
4. The predicted value for each terminal node (leaf) is the mean of the target variable in that region.

Mathematical Criterion for Split

For a split based on variable X_j at split point s:

$$R_1(j, s) = \{X | X_j \leq s\}, \quad R_2(j, s) = \{X | X_j > s\}$$

The cost function is:

$$C(j, s) = \sum_{x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2$$

We choose (j^*, s^*) that **minimizes** $C(j, s)$.

Dataset

We want to predict the **Sales (Y)** based on the **Advertising Spend (X)** (in ₹ lakh):

Observation	X (Advertising)	Y (Sales)
1	2	4
2	4	6
3	6	8
4	8	10
5	10	11

We will build a **simple regression tree** using one variable X.

Step 1: Find Possible Splits

The possible split points are midpoints between X values:

$$s = 3, 5, 7, 9$$

Step 2: For each split, compute SSE

We'll calculate for each split:

$$SSE = \sum (y_i - \bar{y}_{left})^2 + \sum (y_i - \bar{y}_{right})^2$$

Split 1: s = 3

Left (X ≤ 3): Y = [4] → mean = 4

Right (X > 3): Y = [6, 8, 10, 11] → mean = 8.75

SSE =

Left: $(4-4)^2 = 0$

Right: $(6-8.75)^2 + (8-8.75)^2 + (10-8.75)^2 + (11-8.75)^2$

$= 7.56 + 0.56 + 1.56 + 5.06 =$ **14.74**

→ **Total SSE = 14.74**

(Variance of Error)

Split 2: s = 5

Left (X ≤ 5): Y = [4, 6] → mean = 5

Right (X > 5): Y = [8, 10, 11] → mean = 9.67

SSE =

Left: $(4-5)^2 + (6-5)^2 = 1 + 1 = 2$

Right: $(8-9.67)^2 + (10-9.67)^2 + (11-9.67)^2 = 2.78 + 0.11 + 1.78 = 4.67$

→ **Total SSE = 2 + 4.67 = 6.67**

Variance

Split 3: s = 7

Left ($X \leq 7$): Y = [4, 6, 8] → mean = 6

Right ($X > 7$): Y = [10, 11] → mean = 10.5

SSE =

Left: $(4-6)^2 + (6-6)^2 + (8-6)^2 = 4 + 0 + 4 = 8$

Right: $(10-10.5)^2 + (11-10.5)^2 = 0.25 + 0.25 = 0.5$

→ Total SSE = 8 + 0.5 = 8.5

Split 4: s = 9

Left ($X \leq 9$): Y = [4, 6, 8, 10] → mean = 7

Right ($X > 9$): Y = [11] → mean = 11

SSE =

Left: $(4-7)^2 + (6-7)^2 + (8-7)^2 + (10-7)^2 = 9 + 1 + 1 + 9 = 20$

Right: $(11-11)^2 = 0$

→ Total SSE = 20

Step 3: Choose the Best Split

Split	SSE	Decision
s = 3	14.74	—
s = 5	**6.67 (minimum)**	✅ Best
s = 7	8.5	—
s = 9	20	—

Hence, the **best split** is at **X = 5**.

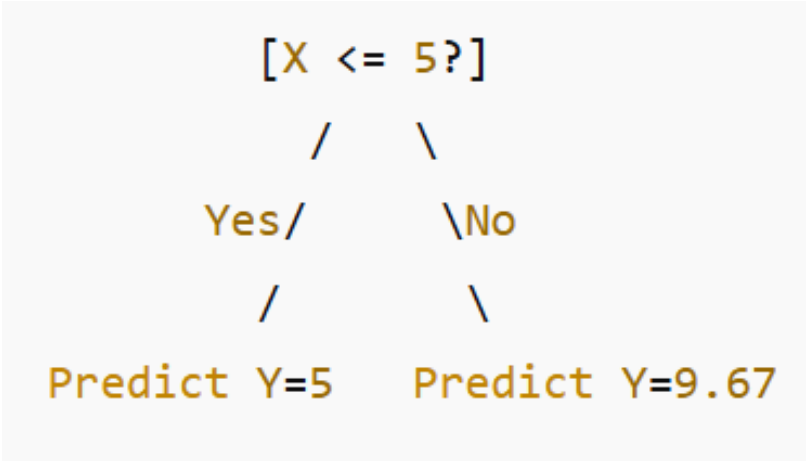
Step 4: Interpret the Tree

- **If X ≤ 5**: predict Y = 5
- **If X > 5**: predict Y = 9.67

Step 5: Predicted values

X	Y (Actual)	Predicted Y
2	4	5
4	6	5
6	8	9.67
8	10	9.67
10	11	9.67

Final Regression Tree



Mean Squared Error (MSE) in Regression Analysis

In regression analysis, the **Mean Squared Error (MSE)** is a commonly used metric to measure how well a regression model fits the data. It represents the **average of the squares of the errors** — that is, the average squared difference between the **actual (observed)** and the **predicted** values. It is used as a **loss function** in regression algorithms (e.g., linear regression, neural networks). During model training, the goal is often to **minimize the MSE**.

If the actual values are $y_1, y_2, \ldots, y_n$ and the corresponding predicted values from the regression model are $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$, then:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where:

- y_i = actual value
- $\hat{y}_i$ = predicted value
- n = number of data points

Interpretation

- **MSE = 0** means a perfect fit (predictions are exactly equal to actual values).
- A **smaller MSE** value indicates a better fit of the regression model to the data.
- A **larger MSE** value indicates poor predictive accuracy — the model's predictions deviate more from actual data.

Observation (i)	Actual Value y_i	Predicted Value $\hat{y}_i$	Error $y_i - \hat{y}_i$	Squared Error $(y_i - \hat{y}_i)^2$
1	10	12	-2	4
2	8	9	-1	1
3	12	11	1	1
4	14	13	1	1

$$\text{MSE} = \frac{4+1+1+1}{4} = \frac{7}{4} = 1.75$$

Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2}$	Square root of MSE; same units as the response variable.
R^2 (Coefficient of Determination)	$1 - \frac{\text{SSR}}{\text{SST}}$	Measures proportion of variance explained by the model.

Advantages of MSE

- Easy to compute and widely used.
- Useful in optimization algorithms (like gradient descent in machine learning).
- Penalizes large errors heavily due to squaring, which helps discourage outliers.

Limitations of MSE

- Sensitive to outliers because errors are squared.
- The value is not in the same units as the original data (since errors are squared).
- May not be intuitive for interpretation compared to MAE.

Usage in Model Evaluation

In regression analysis and machine learning:

- MSE helps compare different models — the one with the lowest MSE is usually preferred.
- It is commonly used as the loss function in training models like **Linear Regression**, **Ridge Regression**, and **Neural Networks**.

Numerical on multiple regression analysis

Problem

A company wants to predict **Sales (Y)** based on two independent variables:

- X_1 = Advertising expenditure (in ₹ lakhs)
- X_2 = Number of salespeople

The following data is collected:

Observation	X_1	X_2	Y (Sales)
1	1	2	2
2	2	1	3
3	3	4	6
4	4	3	7
5	5	5	11

We need to find the regression equation:

$$Y = b_0 + b_1 X_1 + b_2 X_2$$

STEP 1: Compute the required means

$$\bar{X_1} = \frac{1+2+3+4+5}{5} = 3$$

$$\bar{X_2} = \frac{2+1+4+3+5}{5} = 3$$

$$\bar{Y} = \frac{2+3+6+7+11}{5} = 5.8$$

STEP 2: Compute the deviations

Obs	X_1	X_2	Y	$(X_1-\bar{X}_1)$	$(X_2-\bar{X}_2)$	$(Y-\bar{Y})$
1	1	2	2	-2	-1	-3.8
2	2	1	3	-1	-2	-2.8
3	3	4	6	0	1	0.2
4	4	3	7	1	0	1.2
5	5	5	11	2	2	5.2

STEP 3: Compute the necessary sums

Quantity	Formula	Calculation	Value
S_{x1x1}	$\Sigma(X_1-\bar{X}_1)^2$	$(-2)^2 + (-1)^2 + 0^2 + 1^2 + 2^2$	10
S_{x2x2}	$\Sigma(X_2-\bar{X}_2)^2$	$(-1)^2 + (-2)^2 + 1^2 + 0^2 + 2^2$	10
S_{x1x2}	$\Sigma(X_1-\bar{X}_1)(X_2-\bar{X}_2)$	(-2)(-1) + (-1)(-2) + 0(1) + 1(0) + 2(2)	8
S_{x1y}	$\Sigma(X_1-\bar{X}_1)(Y-\bar{Y})$	(-2)(-3.8)+(-1) (-2.8)+0(0.2)+1(1.2)+2(5.2)	21.0
S_{x2y}	$\Sigma(X_2-\bar{X}_2)(Y-\bar{Y})$	(-1)(-3.8)+(-2) (-2.8)+1(0.2)+0(1.2)+2(5.2)	17.0

STEP 4: Formulas for b_1 and b_2

For two independent variables:

$$b_1 = \frac{S_{x2x2}S_{x1y} - S_{x1x2}S_{x2y}}{S_{x1x1}S_{x2x2} - S_{x1x2}^2}$$

$$b_2 = \frac{S_{x1x1}S_{x2y} - S_{x1x2}S_{x1y}}{S_{x1x1}S_{x2x2} - S_{x1x2}^2}$$

Compute the denominator:

$$D = S_{x1x1}S_{x2x2} - S_{x1x2}^2 = (10)(10) - (8)^2 = 100 - 64 = 36$$

Now compute numerators

$$\text{Numerator for } b_1 = (10)(21.0) - (8)(17.0) = 210 - 136 = 74$$

$$\text{Numerator for } b_2 = (10)(17.0) - (8)(21.0) = 170 - 168 = 2$$

Hence b_1 = 74/36 = 2.056, b_2 = 2/36 = 0.056

STEP 5: Compute the intercept b_0

$$b_0 = \bar{Y} - b_1\bar{X}_1 - b_2\bar{X}_2$$

Value of b_0 is = 5.8 – (2.056 *3) – (0.056 * 3) = = 5.8 – 6.336 = - 0.536

Final Regression Equation is Y = - 0.536 + 2.056 X_1 + 0.056 X_2

Interpretation

- When both X_1 and X_2 = 0, predicted sales = **– 0.536** (the intercept).
- For every unit increase in **advertising expenditure (X_1)**, sales increase by 2.056 **units** (holding X_2 constant).
- For every additional **salesperson (X_2)**, sales increase by **0.667 units** (holding X_1 constant).

Clustering in Optimization

Clustering is an **unsupervised technique** used to group a set of objects into clusters such that:

- Objects within a cluster are **similar to each other**, and
- Objects in different clusters are **dissimilar**.

Its primary goal is to identify hidden patterns and segment large datasets into smaller, meaningful groups for business applications like customer segmentation, risk management, anomaly detection, customer behaviour analysis, strategic decision making in business and pattern recognition.

Objectives of Clustering

1. Identify hidden patterns in data.
2. Discover customer segments or behavior groups.
3. Reduce data complexity by summarizing large datasets.
4. Support decision-making, marketing, and risk management.

Applications of Clustering in Optimization Methods

Clustering is the process of grouping similar data points together based on distance or similarity measures. In optimization, clustering is widely used because it reduces computational effort, improves model accuracy, and helps structure complex optimization problems.

1. Reducing Problem Size (Dimensionality Reduction)

Clustering helps **break a large optimization problem into smaller subproblems**.
For example:

- Grouping customers in transportation or vehicle routing problems.
- Solving each cluster separately reduces total computation time.

2. Initialization of Optimization Algorithms

Many optimization algorithms (like *k-means, gradient-based search*) depend heavily on good initial points. Clustering provides:

- **Representative centroids** that serve as high-quality initial guesses.
- Faster convergence and fewer iterations.

3. Data Preprocessing in Machine Learning Optimization

In machine learning, optimization is used to train models. Clustering helps by:

- Detecting **outliers** that may distort the optimization landscape.
- Creating **mini-batches** for stochastic gradient descent (SGD).
- Grouping similar samples to speed up learning.

4. Solving Facility Location and Logistics Optimization

In operations research problems:

- Clustering groups customers/locations to find optimal warehouse or hub positions.
- k-means is directly used to determine:
    - Best facility locations
    - Distribution centers
    - Regions served by each facility

This significantly reduces routing and transportation costs.

5. Feature Engineering for Optimization Models

Clustering creates high-level features that:

- Simplify complex optimization models,
- Improve accuracy of objective functions,
- Enhance performance of predictive optimization techniques.

6. Market Segmentation in Decision Optimization

Organizations use clustering to segment:

- Customers
- Products
- Behaviors

These segments feed into:

- Pricing optimization
- Resource allocation
- Demand forecasting

Improving decision-making quality.

Clustering supports optimization by:

- Reducing problem size
- Providing good starting points
- Identifying structure in the search space
- Improving convergence in iterative algorithms
- Supporting logistics, facility location, and machine learning models

Methods of Clustering

There are several clustering approaches, each based on how the similarity between data points is defined and how clusters are formed.

a) Partitioning Methods

These methods divide data into **k non-overlapping clusters** directly.
Each cluster is represented by a **centroid** (mean point).

Algorithm: *K-Means Clustering*

- Choose number of clusters ***k***.
- Initialize ***k*** cluster centers (randomly).
- Assign each data point to the nearest centroid (using distance measure like Euclidean distance).
- Recompute centroids as the mean of points in each cluster.
- Repeat until centroids stabilize.

Example: A retail company uses K-Means to group 1,000 customers into **3 segments**:

- Cluster 1: Price-sensitive customers
- Cluster 2: Brand-loyal customers
- Cluster 3: Occasional buyers

b) Hierarchical Clustering Methods

Builds a hierarchy (tree structure) of clusters, known as a **dendrogram**.

Example:
In customer data, hierarchical clustering may reveal:

- Level 1: High-value vs. low-value customers
- Level 2: Within high-value → online vs. in-store buyers

Question: A center records customer data (Age, BMI) to classify new members

Person	Age	BMI	Category
P1	25	22	Fit
P2	30	28	Unfit
P3	45	25	Fit
P4	50	30	Unfit
P5	35	26	Fit

a) A new member P6 has Age =40, BMI=27. Use **k = 3** to classify the new member?

b) Perform **k-means clustering (k = 2)** on the same dataset (Age, BMI) for two iterations taking initial centroids as (25,22) and (45,25)? Show the initial centroids, cluster assignment, and updated centroids after each iteration of the k-means clustering algorithm applied in this problem?

SOLUTION

New member: P6 → (Age = 40, BMI = 27)

We use **Euclidean distance**:

Person	Distance from P6
P1	$\sqrt{((40-25)^2 + (27-22)^2)} = \sqrt{250} =$ **15.81**
P2	$\sqrt{((40-30)^2 + (27-28)^2)} = \sqrt{101} =$ **10.05**
P3	$\sqrt{((40-45)^2 + (27-25)^2)} = \sqrt{29} =$ **5.39**
P4	$\sqrt{((40-50)^2 + (27-30)^2)} = \sqrt{109} =$ **10.44**
P5	$\sqrt{((40-35)^2 + (27-26)^2)} = \sqrt{26} =$ **5.10**

3 Nearest Neighbours are P5 (Fit), P3 (Fit), P2 (Unfit)

Majority class = Fit. So, P6 is classified as → Fit.

(b) K-Means Clustering (k = 2)

Initial Centroids are **C1 = (25, 22)** (P1) and **C2 = (45, 25)** (P3).

Distance & Cluster Assignment

Person	Distance to C1	Distance to C2	Cluster
P1 (25,22)	0	20.22	C1
P2 (30,28)	7.81	15.13	C1
P3 (45,25)	20.22	0	C2
P4 (50,30)	32.02	7.07	C2
P5 (35,26)	14.87	10.05	C2

Clusters after Iteration 1 : Cluster C1: P1, P2 Cluster C2: P3, P4, P5

Updated Centroids

New C1

$$\left(\frac{25+30}{2}, \frac{22+28}{2}\right) = (27.5, 25)$$

New C2

$$\left(\frac{45+50+35}{3}, \frac{25+30+26}{3}\right) = (43.33, 27)$$

Iteration 2

Person	Distance to New C1	Distance to New C2	Cluster
P1	4.30	18.42	C1
P2	3.90	13.33	C1
P3	17.68	2.40	C2
P4	29.18	7.64	C2
P5	7.57	8.33	C1

Clusters after Iteration 2 are : Cluster C1: P1, P2, P5 Cluster C2: P3, P4

Updated Centroids

Final C1

$$\left(\frac{25+30+35}{3}, \frac{22+28+26}{3}\right) = (30, 25.33)$$

Final C2

$$\left(\frac{45+50}{2}, \frac{25+30}{2}\right) = (47.5, 27.5)$$

Practice Question on Linear Regression Analysis

Quarterly data for the past 2 years is given. Evaluate the trend forecast for each quarter of the upcoming year (year 3) using linear regression [10][CO4][BTL5]

Period	1	2	3	4	5	6	7	8
Demand	569	564.1	578.9	587.8	789	793.9	779.1	770.2

Q5 b) Step1: Linear Regression Model is: Y = a + bt or Y= a + bX

$$b = \frac{\sum(t - \bar{t})(Y - \bar{Y})}{\sum(t - \bar{t})^2}$$

$$a = \bar{Y} - b\bar{t}$$

The means of X(Period t) and Y (Demand) are 4.5 and 679.

The regression coefficients are **Slope (b) = 39.64, Intercept (a) = 500.63**

So, the linear Regression equation is **Y = 500.63 + 39.64 X**

Step 3: Forecast for Year 3 (Periods 9 to 12)

Quarter (t)	Forecast Demand
9 (Q1)	500.63 + 39.64 x 9 = **857.37**
10 (Q2)	500.63 + 39.64 x 10 = **897.01**
11 (Q3)	500.63 + 39.64 x 11 = **936.65**
12 (Q4)	500.63 + 39.64 x 12 = **976.29**

Q. Explain the condition in optimization when a Linear Programming Problem has an unbounded solution? use an example to justify your answe

This occurs when:

An LPP is unbounded when the feasible region is open in the direction of optimization, and the objective function can increase (or decrease) indefinitely without violating any constraints. Meaning:

- Constraints do not restrict the feasible region in the direction of optimization.
- As a result, the value of the objective function can grow infinitely.

In Simplex Method, if all the coefficients in the entering column are ≤ 0,

i.e., All entries are zero or negative, and therefore, no positive denominator exists for the ratio test then no leaving variable can be selected. This means the entering variable can increase indefinitely without violating any constraint.

So, this is the case of an unbounded solution of a LPP.

Example is as follows:

Maximize $Z=3x+4y$

Subject to constraints: $x-y\geq 2,\ x,y\geq 0$

Newton's Method in Optimization

Newton's Method (or the Newton–Raphson method) is a **second-order iterative optimization technique** used to find **stationary points** (minima, maxima, or saddle points) of a real-valued differentiable function $f(x)$.

It extends the 1D Newton–Raphson root-finding method to optimization problems by finding where the **gradient** (first derivative) becomes zero.

Objective:

We want to find x^* such that:

$$\nabla f(x^*) = 0$$

where

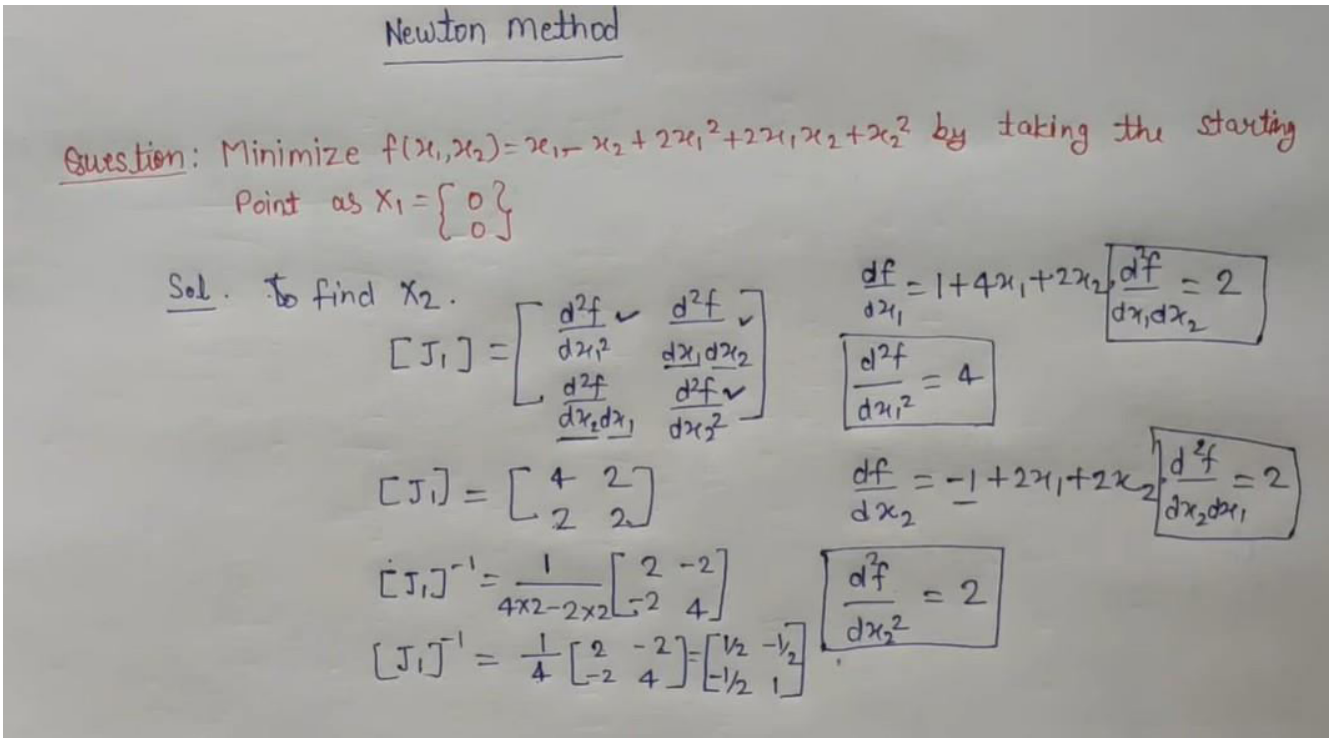
- $\nabla f(x)$ = gradient vector of $f(x)$,
- $\nabla^2 f(x)$ = Hessian matrix (matrix of second derivatives).

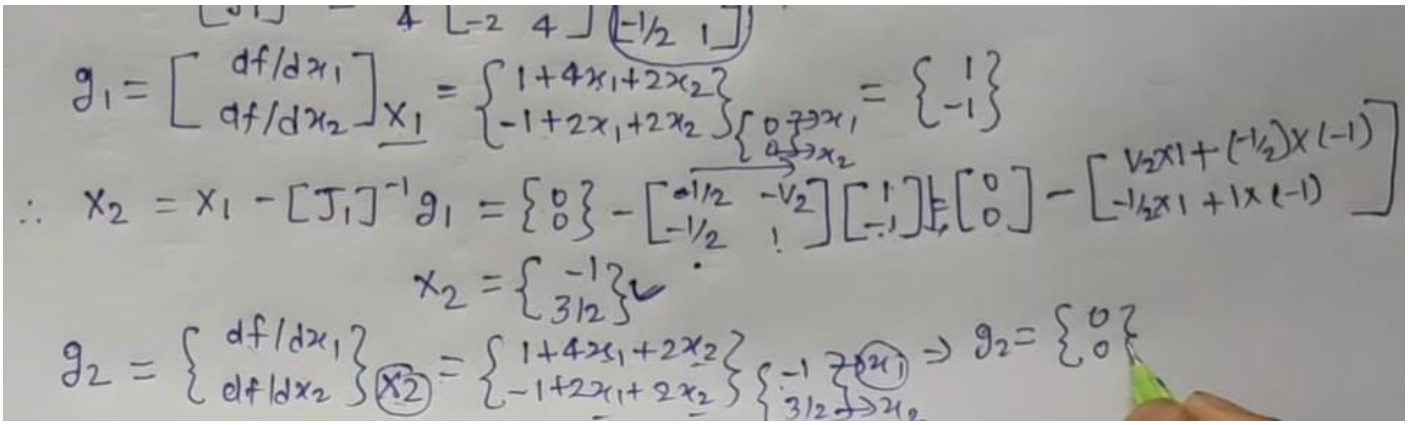
Algorithm

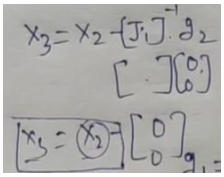
1. **Initialize:** Choose a starting point x_0.
2. **Compute gradient:** $g_k = \nabla f(x_k)$.
3. **Compute Hessian:** $H_k = \nabla^2 f(x_k)$.
4. **Compute search direction:** $d_k = -H_k^{-1} g_k$.
5. **Update:** $x_{k+1} = x_k + d_k$.
6. **Check convergence:** If $||g_{k+1}|| < \epsilon$, stop; else repeat.

Disadvantages of Newton's Method

Disadvantage	Explanation
Requires Hessian computation	The Hessian ((n \times n) matrix) must be computed and inverted — expensive for large (n).
May not converge	If the Hessian is not positive definite (saddle point or maximum), the step can move away from minimum.
Sensitive to initial guess	Poor starting point can lead to divergence or convergence to the wrong stationary point.
High computational cost	Computing and inverting the Hessian costs (O(n^3)).
Not suitable for non-smooth functions	Requires continuous second derivatives.
Step may overshoot	If the step size is too large, the quadratic approximation fails — often a line search or damping factor is added.





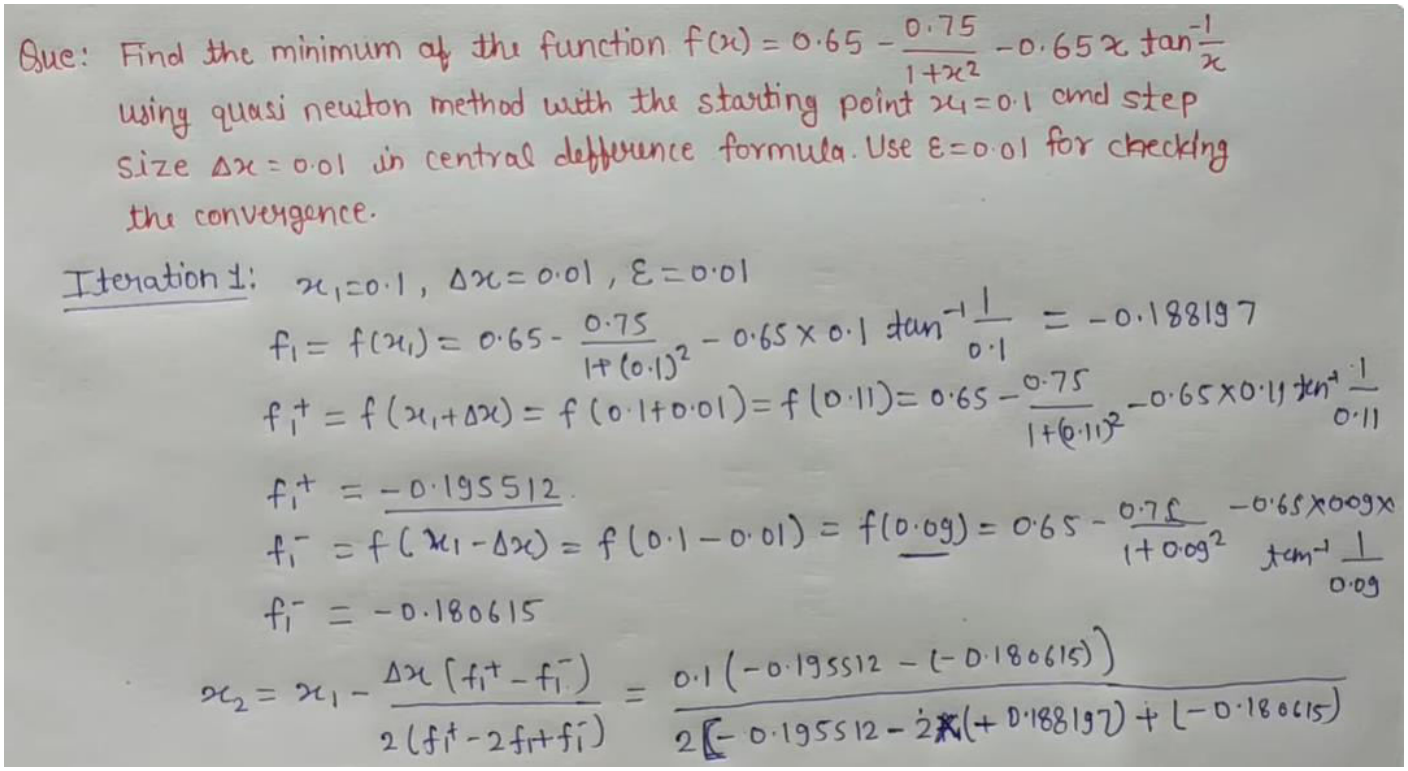


Newton vs. Gradient Descent

Feature	Gradient Descent	Newton's Method
Uses	Gradient only	Gradient + Hessian
Step Direction	Negative gradient	Inverse Hessian × gradient
Convergence Rate	Linear	Quadratic (faster near optimum)
Cost per Iteration	Low	High (Hessian inversion)
Suitable for	Large-scale, simple problems	Smaller, well-behaved quadratic problems

Quasi Newton Method

- Quasi-Newton methods are optimization algorithms that find minima or maxima of functions by approximating the Hessian matrix of second derivatives.
- Unlike Newton's method, they avoid the computational cost of calculating the true Hessian, instead updating an approximation using gradient and position information from previous steps.
- This makes them more efficient for large-scale problems while still achieving superlinear convergence.



So, we have $x_2 = 0.377882$

$$|f'(x_2)| = \left|\frac{f_2^+ - f_2^-}{2\Delta x}\right| = 0.137300 > \varepsilon = 0.01$$

$f_2^+ = f(x_2 + \Delta x)$
$= f(0.377882 + 0.01)$
$= f(0.387882)$
$= -0.304662$

$f_2^- = f(x_2 - \Delta x)$
$= f(0.377882 - 0.01)$
$= f(0.367882)$
$= -0.301916$

Iteration 2:

$f_2 = f(x_2) = -0.303368.$

$f_2^+ = 0.304662$

$f_2^- = -0.301916.$

$$x_3 = x_2 - \frac{\Delta x(f_2^+ - f_2^-)}{2(f_2^+ - 2f_2 + 2f_2^-)} = 0.465390$$

Convergence check.

$$|f'(x_3)| = \left|\frac{f_3^+ - f_3^-}{2\Delta x}\right| = 0.0177 > \varepsilon = 0.01$$

$f_3^+ = f(x_3 + \Delta x)$
$= f(0.465390 + 0.01)$
$= f(0.475390)$
$= -0.310004$

$f_3^- = f(x_3 - \Delta x)$
$= f(0.465390)$
$= -0.455390$

Iteration 3:

$$f_3 = f(x_3) = 0.65 - \frac{0.75}{1 + 0.46539^2} - 0.65 \times 0.46539 \times \tan^{-1}\frac{1}{0.46539}$$

$f_3 = -0.309885$

$f_3^+ = -0.310004, \quad f_3^- = -0.309650$

$$x_4 = x_3 - \frac{\Delta x(f_3^+ - f_3^-)}{2(f_3^+ - 2f_3 + f_3^-)} = 0.480600$$

Convergence check.

$$|f'(x_4)| = \left|\frac{f_4^+ - f_4^-}{2\Delta x}\right| = 0.000350 < \varepsilon = 0.01$$

$f_4^+ = f(x_4 + \Delta x)$
$= f(0.490600)$
$= -0.3099688$

$f_4^- = f(x_4 - \Delta x)$
$= f(0.47060)$
$= -0.3099615$

process has converged we take the optimum solution as $x^* \approx x_4 = 0.480600.$

Association Rule in Data Mining

Association Rule Mining is a data mining technique used to find relationships or correlations among items in large transactional databases. It helps answer questions like:

"If a customer buys item X, how likely are they to buy item Y?"

This technique is most famously used in **market basket analysis** — e.g., finding that *"70% of customers who buy bread also buy butter."*

Structure of an Association Rule

A rule is generally written as: $X \rightarrow Y$ where **X** and **Y** are itemsets. **X** denotes the antecedent (if-part) and **Y** denotes the consequent (then-part)

Example: Bread $\rightarrow$ Butter means customers who buy Bread are likely to buy Butter.

Association Rule Metrics

To evaluate the strength of association rules, three key measures are used:

(a) Support - Support shows how frequently the rule occurs in the dataset.

$$\text{Support}(X \Rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Total number of transactions}}$$

Support tells how popular (frequency of both items) the rule is in the overall dataset.

(b) Confidence - shows how often items in Y appear in transactions that contain X.

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Transactions containing X}}$$

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

It measures the **reliability** of the inference made by the rule.

(c) Lift - measures how much more likely Y is bought when X is bought.

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

It is the ratio of the observed support to that expected if X and Y were expected (measures the degree of corelation between Xand Y).

If **Lift > 1** → positive correlation between X and Y (items occur together more than expected)

If **Lift = 1** → X and Y are independent.

If **Lift < 1** → negative correlation.

Numerical Problem- A store has the following transaction data:

Transaction ID	Items Purchased
T1	Milk, Bread, Butter
T2	Bread, Butter
T3	Milk, Bread
T4	Milk, Bread, Butter
T5	Bread, Butter

Find the **Support**, **Confidence**, and **Lift** for the rule: Bread → Butter

SOLUTION **Step 1: Total Transactions**
Total = **5**

Step 2: Count occurrences

- Transactions containing **Bread** = T1, T2, T3, T4, T5 → **5**
- Transactions containing **Butter** = T1, T2, T4, T5 → **4**
- Transactions containing **Bread and Butter** = T1, T2, T4, T5 → **4**

Step 3: Compute Metrics

(a) Support of Rule

$$\text{Support}(Bread \Rightarrow Butter) = \frac{4}{5} = 0.8$$

→ **80% of all transactions** contain both Bread and Butter.

(b) Confidence of Rule

$$\text{Confidence}(Bread \Rightarrow Butter) = \frac{4}{5} = 0.8$$

→ When Bread is bought, Butter is also bought **80%** of the time.

(c) Lift of Rule

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

$$\text{Lift}(Bread \Rightarrow Butter) = \frac{0.8}{\text{Support}(\text{Butter})} = \frac{0.8}{0.8} = 1.0$$

Since **Lift = 1**, Bread and Butter are **independent** — buying Bread does not particularly increase or decrease the likelihood of buying Butter.

5. Interpretation

- **High Support (0.8)** → Rule is common.
- **High Confidence (0.8)** → Rule is reliable.
- **Lift = 1.0** → No extra influence; items are independent.

Problem 2: Multi-Item Antecedent Rule

TID	Items Purchased
T1	Milk, Bread, Butter, Eggs
T2	Bread, Butter
T3	Milk, Bread, Butter
T4	Bread, Eggs
T5	Milk, Bread, Eggs

Find Support, Confidence, and Lift for the rule (Milk, Bread) → Butter

Solution

- Total transactions = **5**
- Transactions with (Milk & Bread) = T1, T3, T5 → **3**
- Transactions with (Milk, Bread & Butter) = T1, T3 → **2**
- Transactions with (Butter) = T1, T2, T3 → **3**

$$\text{Support}(X \Rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Total number of transactions}}$$

Support= Transactions with (Milk, Bread & Butter) ÷ Total transactions = 2/5= **0.4(40%)**

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Transactions containing X}}$$

Confidence = 2/3 = 0.667 (67%)

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

Lift = 2/3 ÷ (3/5) = **10/9** = **1.11 which is > 1, implies a positive association.**

Interpretation: When both Milk and Bread are bought, customers are 11% more likely to buy Butter than average → slight positive association.

Problem 3: Complex Case — Multi-item Consequent

TID	Items Purchased
T1	Bread, Butter, Milk, Eggs
T2	Bread, Milk
T3	Bread, Butter
T4	Milk, Eggs
T5	Bread, Butter, Eggs

Find Support, Confidence, and Lift for the rule Milk → (Bread, Butter)

Solution

- Total transactions = 5
- Transactions with Milk = T1, T2, T4 → 3
- Transactions with Bread & Butter = T1, T3, T5 → 3
- Transactions with Milk, Bread & Butter = T1 → 1

Support = 1/5 = 0.2(20%)

Confidence = 1/3 = 0.333

Lift = 1/3 ÷ 3/5 = **5/9 = 0.555** which is < 1

Interpretation: Lift < 1 → Customers who buy Milk are less likely to buy both Bread and Butter together compared to the general population.

Problem 4: Comparing Two Rules

TID	Items Purchased
T1	Pen, Notebook, Eraser
T2	Pen, Notebook
T3	Pen, Eraser
T4	Notebook, Pencil
T5	Pen, Notebook, Pencil

Compare rules: Pen $\rightarrow$ Notebook **and** Notebook $\rightarrow$ Pen.

Solution : The total transactions = 5

Transactions with Pen = T1, T2, T3, T5 → 4

Transactions with Notebook = T1, T2, T4, T5 → 4

Transactions with both = T1, T2, T5 → 3

For Rule 1 : Pen $\rightarrow$ Notebook

Support = 3/5 = 0.6
Confidence = 3/4 = 0.75
Lift = 0.75 / 0.8 = 0.9375

For Rule 2 : Notebook $\rightarrow$ Pen

Support = 0.6
Confidence = 3/4 = 0.75
Lift = 0.75 / 0.8 = 0.9375

Interpretation: Both rules are symmetric, and Lift < 1 → **slightly negative correlation**

Apriori Algorithm in Data Mining

It is a basic method used in DM to find groups of items that often appear together in large sets of data. It helps to discover useful patterns or rules about how items are related which is particularly valuable in market basket analysis. *Like in a grocery store if many customers buy bread and butter together, the store can use this information to place these items closer or create special offers. This helps the store sell more and make customers happy*

This **algorithm** is one of the **most fundamental algorithms** in data mining, specifically used for **association rule learning** — to find frequent itemsets and derive meaningful **association rules** (like *"Customers who buy bread also buy butter"*).

It is based on the principle that:

"All subsets of a frequent itemset must also be frequent."

This principle is called the **Apriori property.**

Objective: To find

- All **frequent itemsets** that occur above a given **minimum support threshold**.
- Strong **association rules** that have **confidence** above a given **minimum confidence threshold**.

Key Concepts and Formulas

Term	Definition	Formula
Support (A ⇒ B)	Frequency of transactions containing both A and B	$\text{Support}(A \Rightarrow B) = \frac{n(A \cup B)}{N}$
Confidence (A ⇒ B)	Likelihood of buying B when A is bought	$\text{Confidence}(A \Rightarrow B) = \frac{n(A \cup B)}{n(A)}$
Lift (A ⇒ B)	Strength of rule compared to random co-occurrence	$\text{Lift}(A \Rightarrow B) = \frac{\text{Confidence}(A \Rightarrow B)}{\text{Support}(B)}$

Steps of the Apriori Algorithm

1. **Set minimum support (min_sup)** and **minimum confidence (min_conf)** thresholds.
2. **Generate all 1-itemsets** and count their frequency.
3. **Prune** itemsets that do not meet min_sup.
4. Use frequent itemsets of size k to generate **candidate itemsets of size (k+1)** (self-join).
5. **Count frequency** of these candidates and again prune those not meeting min_sup.
6. Continue until no more frequent itemsets can be generated.
7. From the frequent itemsets, **generate strong association rules** using the min_conf threshold.

Numerical Example

TID	Items Purchased
T1	Milk, Bread, Butter
T2	Bread, Butter
T3	Milk, Bread
T4	Milk, Bread, Butter
T5	Bread, Butter, Eggs

Given: Min. Support = 60% (i.e., 0.6), Min. Confidence = 70% (i.e., 0.7) and transactions = 5

Step 1 : Generate 1-itemsets

Item	Count	Support
Milk	3	3/5 = 0.6
Bread	5	5/5 = 1.0
Butter	4	4/5 = 0.8
Eggs	1	1/5 = 0.2

Step 2: Generate 2-itemsets

Possible pairs: (Milk, Bread), (Milk, Butter), (Bread, Butter)

Itemset	Count	Support
(Milk, Bread)	3	0.6
(Milk, Butter)	2	0.4
(Bread, Butter)	4	0.8

Frequent 2-itemsets: (Milk, Bread), (Bread, Butter) as they meet minimum support criteria

Step 3: Generate 3-itemsets

Possible 3-itemset: (Milk, Bread, Butter)

Itemset	Count	Support
(Milk, Bread, Butter)	2	0.4

Not frequent (Support = 0.4 < 0.6)

Final frequent itemsets: {Milk}, {Bread}, {Butter}, {Milk, Bread}, {Bread, Butter}

Step 4: Generate Association Rules

From **2-itemsets**, we can derive:

Rule 1: Milk ⇒ Bread

Support = 0.6

Confidence = (3/3) = 1.0 ✅

Lift = 1.0 / 1.0 = 1.0

Rule 2: Bread ⇒ Milk

Support = 0.6

Confidence = (3/5) = 0.6 ❌ (below 0.7 threshold)

Rule 3: Bread ⇒ Butter

Support = 0.8

Confidence = (4/5) = 0.8 ✅

Lift = 0.8 / 0.8 = 1.0

Rule 4: Butter ⇒ Bread

Support = 0.8

Confidence = (4/4) = 1.0 ✅

Lift = 1.0 / 1.0 = 1.0

Strong Rules (min_conf ≥ 0.7):

Milk ⇒ Bread (Confidence = 1.0)

Bread ⇒ Butter (Confidence = 0.8)

Butter ⇒ Bread (Confidence = 1.0)

Interpretation

- Milk ⇒ Bread → Every customer who buys Milk also buys Bread.
- Butter ⇒ Bread → Butter sales are always accompanied by Bread purchases.
- Bread ⇒ Butter → Most Bread buyers also buy Butter (80% of the time).

These insights help in cross-selling and product placement strategies.

Summary table

Rule	Support	Confidence	Lift	Strength
Milk ⇒ Bread	0.6	1.0	1.0	Strong
Bread ⇒ Butter	0.8	0.8	1.0	Strong
Butter ⇒ Bread	0.8	1.0	1.0	Strong

- **Apriori** uses the “subset” property to reduce computation.
- It is **bottom-up** — starting with 1-itemsets and building up.
- Performance depends on **support/confidence thresholds**.

REGRESSION ANALYSIS IN DATA MINING

Regression Analysis is a **predictive modeling technique** used to understand the relationship between a **dependent variable (target)** and one or more **independent variables (predictors)**. In **data mining**, regression is used to **forecast trends**, **predict numeric outcomes**, and **analyze relationships** between variables — for example:

- Predicting sales revenue based on advertising expenditure.
- Estimating house prices based on size, location, and number of rooms.
- Forecasting stock prices based on past data.

Purpose of Regression Analysis in Data Mining

Regression is part of **predictive analytics**, and it helps in:

- **Prediction:** Estimating future or unknown values.
- **Trend Analysis:** Understanding how variables change over time.
- **Relationship Identification:** Measuring how strongly variables are related.
- **Optimization:** Finding values that maximize or minimize outcomes.

Types of Regression Used in Data Mining

Type	Description	Example
Linear Regression	Relationship between variables is linear (straight-line).	Predicting sales based on advertising spend.
Multiple Linear Regression	More than one independent variable.	Predicting house prices based on area, rooms, location.
Logistic Regression	Used when the dependent variable is categorical (e.g., Yes/No).	Predicting if a customer will buy a product.
Polynomial Regression	Non-linear relationships fitted by higher-order equations.	Predicting growth rates or curved trends.

Linear Regression Model

The simplest regression is the **Simple Linear Regression**, represented as: $Y=a + bX + \varepsilon$

Where Y is the dependent variable (Predicted variable), X is the independent variable (Predictor), a is the Intercept (value of Y when X = 0), b is the regression coefficient or slope of the line and ε denotes Error term (unexplained variation) which is assumed as zero. So, the regression line of Y on X is written as $Y=a + bX$, where a, b are to be determined using the dataset values of (X,Y).

Numerical Problem:

A marketing manager wants to understand how **advertising expenditure (X)** affects **sales revenue (Y)**. The data collected from 5 months is:

Month	Advertising (X in ₹'000)	Sales (Y in ₹'000)
1	2	4
2	3	5
3	5	7
4	7	10
5	9	15

Find the **regression equation** and **predict sales** when advertising = ₹6,000.

SOLUTION : Step 1: Compute Required Values

X	Y	X^2	XY
2	4	4	8
3	5	9	15
5	7	25	35
7	10	49	70
9	15	81	135

$$\sum X = 26, \quad \sum Y = 41, \quad \sum X^2 = 168, \quad \sum XY = 263, \quad n = 5$$

Step 2: Find the Slope (b)

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2}$$

$$b = \frac{5(263) - (26)(41)}{5(168) - (26)^2}$$

$$b = \frac{1315 - 1066}{840 - 676} = \frac{249}{164} = 1.518$$

Step 3: Find the Intercept (a)

$$a = \frac{\sum Y - b(\sum X)}{n}$$

$$a = \frac{41 - 1.518(26)}{5} = \frac{41 - 39.468}{5} = 0.306$$

Step 4: Regression Equation

$$Y = a + bX = 0.306 + 1.518X$$

Step 5: Prediction

For $X = 6$ (₹6,000 advertising spend):

$$Y = 0.306 + 1.518(6) = 0.306 + 9.108 = 9.414$$

Predicted Sales = ₹9.41 thousand = ₹9410

Interpretation

1. **Slope (b = 1.518):** For every ₹1,000 increase in advertising, sales increase by ₹1.518 thousand.
2. **Intercept (a = 0.306):** When advertising is zero, sales are expected to be ₹0.306 thousand.
3. Thus, there is a **strong positive linear relationship** between advertising and sales.

Uses of Regression Analysis in Data Mining

Area	Application
Marketing	Predicting sales, demand forecasting
Finance	Credit scoring, stock price prediction
Operations	Resource optimization, capacity planning
HR Analytics	Salary prediction, performance analysis
Healthcare	Predicting patient outcomes or treatment costs

CONCLUSION:

- Regression is a **supervised learning technique** used for **numeric prediction**.
- The **strength** of prediction depends on correlation between variables.
- It is a foundational method for **predictive analytics**, **trend forecasting**, and **decision support systems**.

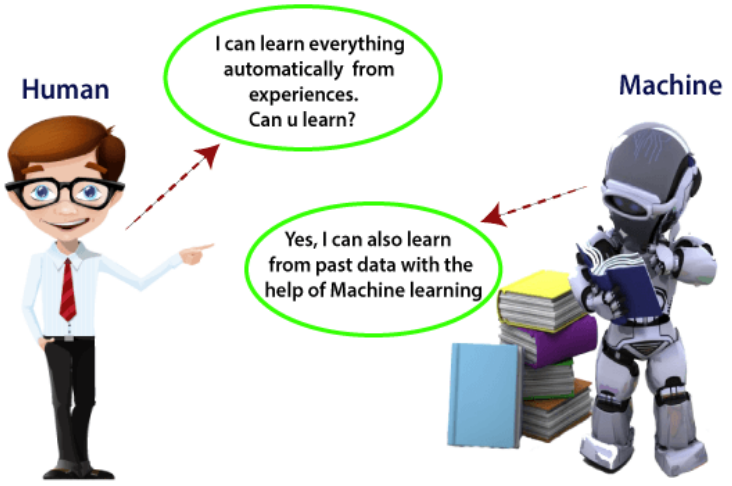
PROBLEM 2 ON REGRESSION

x	y	x^2	xy
8	11	64	88
5	10	25	50
4	4	16	16
6	8	36	48
7	9	49	63
9	13	81	117
10	15	100	150
3	6	9	18
2	12	4	24
12	7	144	84

From the above table

- n=10 , ∑ x = 66 , ∑ y = 95 , ∑ xy =1186 , ∑ x^2 = 528
- Now the normal equations become :
- 95 = 10*b + 66a
- 1186 = 66*b + 528a
- By solving the above two euations we get a = 6.05 and b = -30.429
- The linear regression equation is y = -30.429 + 6.05 x.

Machine Learning - Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for **building mathematical models and making predictions using historical data or information**. Currently, it is being used for various tasks such as **image recognition**, **speech recognition**, **email filtering**, **Facebook auto-tagging**, **recommender system**, and many more.



Machine Learning is said as a subset of AI that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. Arthur Samuel, an early American leader in the field of computer gaming and artificial intelligence, coined the term "Machine Learning " in 1959 while at IBM. He defined machine learning as "**the field of study that gives computers the ability to learn without being explicitly programmed** ". However, there is no universally accepted definition for machine learning

Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

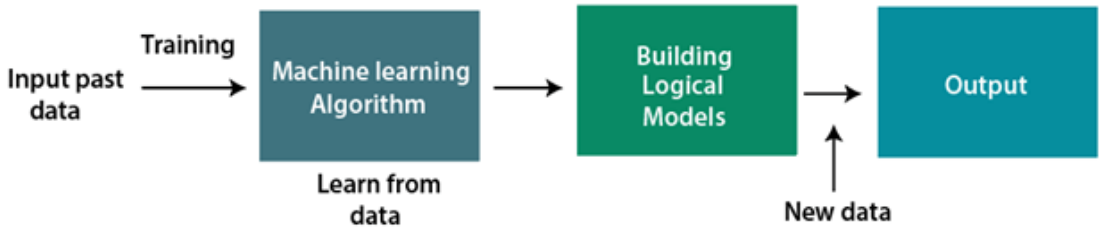
There are several types of machine learning, including **supervised learning, unsupervised learning, and reinforcement learning.**

How does Machine Learning work

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem.

The below block diagram explains the working of Machine Learning algorithm:



Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- ML is much similar to data mining as it also deals with the huge amount of the data.

Need for Machine Learning

The need for ML is increasing day by day. The reason behind the need for ML is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

We can train ML algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of ML can be easily understood by its use-cases. Currently, ML is used in self-driving cars, cyber fraud detection, face recognition, and friend suggestion by Facebook, etc. Various top companies such as Netflix and Amazon have built ML models that are using a vast amount of data to analyze the user interest and recommend product accordingly.

Following are some key points which show the importance of ML :

- Rapid increment in the production of data
- Solving complex problems, which are difficult for a human
- Decision making in various sector including finance
- Finding hidden patterns and extracting useful information from data.

Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. **Supervised learning**
2. **Unsupervised learning**
3. **Reinforcement learning**

Supervised Learning

Supervised learning is a type of ML method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering. Supervised learning can be grouped further in two categories of algorithms:

- **Classification**
- **Regression**

Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. There are two categories of USL algorithms:

- **Clustering**
- **Association**

Reinforcement Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action.

The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

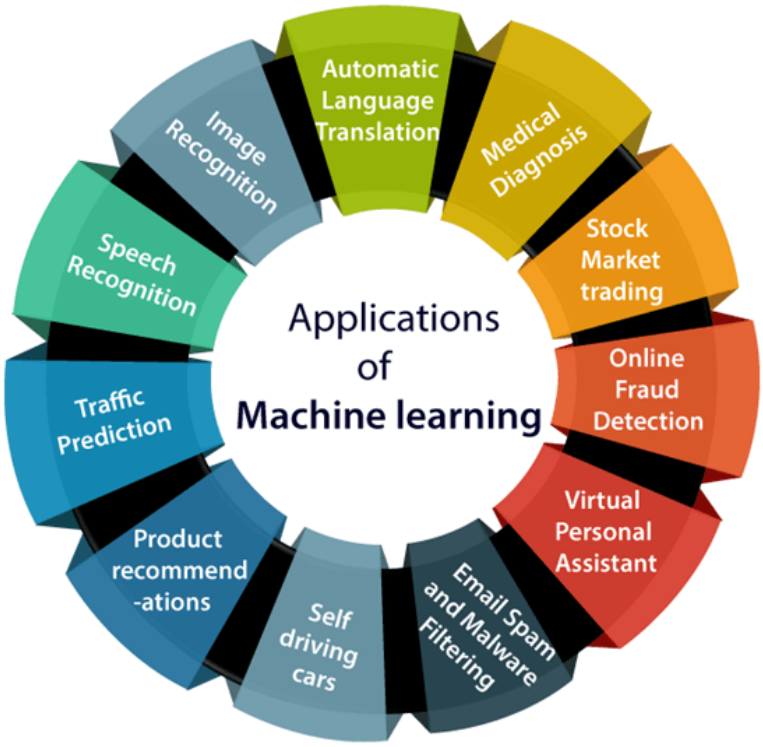
Machine Learning at present:

Now ML has got a great advancement in its research, and it is present everywhere around us, such as self-driving cars, Amazon Alexa, Catboats, recommender system, and many more.

It includes Supervised, unsupervised and reinforcement learning with,
clustering, classification, decision tree, SVM algorithms, etc. Modern machine learning models can be used for making various predictions including **weather prediction, disease prediction, stock market analysis, etc.**

Applications of Machine learning

ML is a buzzword for today's technology, and it is growing very rapidly day by day. We are using ML in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Below are some most trending real-world applications of Machine Learning:



1. Image Recognition:

Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, **Automatic friend tagging suggestion**:

Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's **face detection** and **recognition algorithm**.

It is based on the Facebook project named "**Deep Face**," which is responsible for face recognition and person identification in the picture.

2. Speech Recognition

While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning.

Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**." At present, machine learning algorithms are widely used by various applications of speech recognition. **Google assistant**, **Siri**, **Cortana**, and **Alexa** are using speech recognition technology to follow the voice instructions.

3. Traffic prediction:

If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.

It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:

- **Real Time location** of the vehicle form Google Map app and sensors
- **Average time has taken** on past days at the same time.

Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

4. Product recommendations:

Machine learning is widely used by various e-commerce and entertainment companies such as **Amazon**, **Netflix**, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.

Google understands the user interest using various ML algorithms and suggests the product as per customer interest.As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

5. Self-driving cars:

One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

6. Email Spam and Malware Filtering:

Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Below are some spam filters used by Gmail:

- Content Filter
- Header filter
- General blacklists filter
- Rules-based filters
- Permission filters

Some machine learning algorithms such as **Multi-Layer Perceptron**, **Decision tree**, and **Naïve Bayes classifier** are used for email spam filtering and malware detection.

7. Virtual Personal Assistant:

We have various virtual personal assistants such as **Google assistant**, **Alexa**, **Cortana**, **Siri**. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.

These virtual assistants use machine learning algorithms as an important part.

These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

8. Online Fraud Detection:

Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as **fake accounts**, **fake ids**, and **steal money** in the middle of a transaction. So to detect this, **Feed Forward Neural network** helps us by checking whether it is a genuine transaction or a fraud transaction.

For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.

9. Stock Market trading:

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning's **long short term memory neural network** is used for the prediction of stock market trends.

10. Medical Diagnosis:

In medical science, ML is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain. It helps in finding brain tumors and other brain-related diseases easily.

11. Automatic Language Translation:

Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.

The technology behind the automatic translation is a sequence-to-sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.

UNSUPERVISED ML AND ITS TYPES

1. Clustering: Clustering is the process of grouping similar data points together based on their characteristics or attributes. Clustering algorithms attempt to find patterns in the data and group the data points together based on those patterns.
2. Anomaly detection: Anomaly detection is the process of identifying data points that deviate from the expected behavior of the data set. Anomaly detection algorithms attempt to identify outliers or anomalies in the data.
3. Association rule learning: Association rule learning is the process of discovering relationships between variables in a data set. Association rule learning algorithms attempt to find patterns in the data that indicate that certain variables are associated with each other.
4. Dimensionality reduction: Dimensionality reduction is the process of reducing the number of variables or features in a data set. Dimensionality reduction algorithms attempt to identify the most important features in the data set and remove less important ones.
5. Neural Networks: In unsupervised learning, neural networks can be used for unsupervised representation learning or self-supervised learning. This allows the network to learn useful features of the data without being explicitly given a specific target.

CLUSTERING AND ITS TYPES

K-Means Clustering: It is a popular clustering algorithm that partitions the data into K clusters, where K is a user-defined number. The algorithm works by iteratively assigning data points to their nearest centroid and updating the centroid's position until convergence.

1. Hierarchical Clustering: It is a clustering algorithm that builds a hierarchy of clusters by either merging small clusters into larger ones or splitting large clusters into smaller ones. There are two types of hierarchical clustering: agglomerative (bottom-up) and divisive (top-down).
2. DBSCAN (Density-Based Spatial Clustering of Applications with Noise): It is a clustering algorithm that groups together points that are closely packed together, while marking outliers that lie alone in low-density regions. It works by defining a neighborhood around each data point and grouping points that are densely connected.
3. Mean Shift: It is a non-parametric clustering algorithm that doesn't require specifying the number of clusters beforehand. The algorithm works by iteratively shifting a window (kernel) towards the maximum density of data points until convergence.

4. Spectral Clustering: It is a clustering algorithm that uses the eigenvalues and eigenvectors of a similarity matrix to project the data onto a lower-dimensional space where it can be clustered. The algorithm works by partitioning the graph into two or more clusters based on the eigenvectors.
5. Fuzzy C-Means: It is a soft clustering algorithm that assigns each data point to multiple clusters with varying degrees of membership. The algorithm works by minimizing the sum of the squared distances between the data points and their corresponding cluster centers, weighted by the membership degrees.

REINFORCEMENT LEARNING

Reinforcement learning (RL) is a type of machine learning algorithm that is used for training an agent to make decisions in an environment by interacting with it over time. The agent learns by receiving feedback in the form of rewards or penalties based on its actions, and the goal is to maximize the cumulative reward received over time. Here are the key components of a typical reinforcement learning problem:

Environment: This is the external system or process with which the agent interacts. It can be anything from a simple game environment to a complex real-world scenario such as robotics, traffic control, or finance.

Agent: This is the learner or decision-maker that interacts with the environment. The agent takes actions based on the current state of the environment and receives feedback in the form of rewards or penalties based on the outcomes of its actions.

State: This is a representation of the environment at a given point in time, which the agent uses to make decisions.

Action: This is the decision made by the agent based on the current state of the environment.

Reward: This is the feedback provided to the agent after it takes an action. The reward can be positive or negative and is used to reinforce or discourage certain behaviors.

The reinforcement learning algorithm works by having the agent interact with the environment and learn from its experiences over time. The agent starts by exploring the environment and trying different actions, and the rewards it receives are used to update its policy for taking future actions.

Reinforcement learning is used in a variety of applications, including robotics, game playing, and autonomous vehicles. It is a powerful technique that can learn complex behaviors and can be combined with other machine learning approaches such as deep learning to achieve even better results.

PROBLEM 3: A company's marketing department collected the following data on **advertising expenditure (X)** and **sales revenue (Y)** (both in ₹'000). They want to determine the **regression equation** and **predict sales when advertising = ₹25,000**.

Observation	X (Advertising)	Y (Sales)
1	5	10
2	10	15
3	12	19
4	15	23
5	18	26
6	20	30
7	22	32
8	25	35
9	30	40
10	35	43

Solution:

X	Y	X^2	XY
5	10	25	50
10	15	100	150
12	19	144	228
15	23	225	345
18	26	324	468
20	30	400	600
22	32	484	704
25	35	625	875
30	40	900	1200
35	43	1225	1505

$$\sum X = 212, \quad \sum Y = 273, \quad \sum X^2 = 4452, \quad \sum XY = 6125, \quad n = 10$$

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2}$$

$$b = \frac{10(6125) - (212)(273)}{10(4452) - (212)^2}$$

$$b = \frac{61250 - 57876}{44520 - 44944} = \frac{3374}{576} = 5.86$$

$$a = \frac{\sum Y - b(\sum X)}{n}$$

$$a = \frac{273 - 5.86(212)}{10} = \frac{273 - 1241.32}{10} = \frac{-968.32}{10} = -96.832$$

$$Y = -96.832 + 5.86X$$

Predict sales when **Advertising (X) = 25**: Y = −96.832 + 5.86(25) = 49.668

Problem statement (market-basket)

T1: {Milk, Bread, Butter}

T2: {Bread, Butter, Diapers}

T3: {Milk, Bread, Diapers}

T4: {Milk, Butter, Diapers}

T5: {Bread, Diapers}

T6: {Milk, Bread, Butter}

T7: {Butter, Diapers}

T8: {Milk, Bread}

T9: {Bread, Butter, Diapers}

T10:{Milk, Diapers}

- Minimum support = **30%** (for 10 transactions → min count = 3)
- Minimum confidence = **60%**

Goal: find frequent itemsets and generate association rules meeting min confidence.

SOLUTION TO PROBLEM

Step 1 — Find frequent 1-itemsets (L1)

Count occurrences (support counts and support fraction):

- Milk: appears in T1, T3, T4, T6, T8, T10 → count = **6**, support = 6/10 = **0.60**
- Bread: T1, T2, T3, T5, T6, T8, T9 → count = **7**, support = **0.70**
- Butter: T1, T2, T4, T6, T7, T9 → count = **6**, support = **0.60**
- Diapers: T2, T3, T4, T5, T7, T9, T10 → count = **7**, support = **0.70**

All four items have support ≥ 0.30.

So **L1 = {Milk, Bread, Butter, Diapers}**.

Step 2 — Generate candidate 2-itemsets (C2) and count supports

Candidates (all pairs from L1):

1. {Milk, Bread} — occurs in T1, T3, T6, T8 → count = **4**, support = **0.40**
2. {Milk, Butter} — T1, T4, T6 → count = **3**, support = **0.30**
3. {Milk, Diapers} — T3, T4, T10 → count = **3**, support = **0.30**
4. {Bread, Butter} — T1, T2, T6, T9 → count = **4**, support = **0.40**
5. {Bread, Diapers} — T2, T3, T5, T9 → count = **4**, support = **0.40**
6. {Butter, Diapers} — T2, T4, T7, T9 → count = **4**, support = **0.40**

All candidate pairs meet min support count (≥3).

So **L2 = {MB, MBu, MDi, BBu, BDi, BuDi}** where shorthand: M=Milk, B=Bread, Bu=Butter, Di=Diapers.

Step 3 — Generate candidate 3-itemsets (C3) from L2 and count supports

Possible 3-item combos (from L2 joins):

- {Milk, Bread, Butter} — present in T1, T6 → count = **2**, support = **0.20**
- {Milk, Bread, Diapers} — present in T3 → count = **1**, support = **0.10**
- {Milk, Butter, Diapers} — present in T4 → count = **1**, support = **0.10**
- {Bread, Butter, Diapers} — present in T2, T9 → count = **2**, support = **0.20**

None of these 3-itemsets reach min support count (3). So stop here.

Final frequent itemsets = L1 and L2 (no L3).

Step 4 — Generate association rules from frequent itemsets

We generate candidate rules from frequent 2-itemsets (and could also from larger itemsets, but there are none). For each pair {X,Y} we can form two rules: X ⇒ Y and Y ⇒ X. Use the formulae:

- support(X⇒Y) = count(X ∪ Y) / N
- confidence(X⇒Y) = count(X ∪ Y) / count(X)
- lift(X⇒Y) = confidence(X⇒Y) / support(Y)

We will keep rules with **confidence ≥ 0.60**.

Compute for each frequent pair (counts from above; N = 10):

1. {Milk, Bread} (count = 4, support = 0.40)
    - **Milk ⇒ Bread**
      confidence = 4 / count(Milk=6) = 4/6 = **0.6667** (66.67%) → meets 0.60
      lift = 0.6667 / support(Bread=0.7) = 0.6667 / 0.7 = **0.9524**
    - **Bread ⇒ Milk**
      confidence = 4 / 7 = **0.5714** → below 0.60 (discard)
2. {Milk, Butter} (count = 3, support = 0.30)
    - Milk ⇒ Butter: conf = 3/6 = **0.5** → **discard**
    - Butter ⇒ Milk: conf = 3/6 = **0.5** → **discard**
3. {Milk, Diapers} (count = 3, support = 0.30)
    - Milk ⇒ Diapers: conf = 3/6 = **0.5** → **discard**
    - Diapers ⇒ Milk: conf = 3/7 ≈ **0.4286** → **discard**
4. {Bread, Butter} (count = 4, support = 0.40)
    - Bread ⇒ Butter: conf = 4/7 ≈ **0.5714** → **discard**
    - Butter ⇒ Bread: conf = 4/6 = **0.6667** → meets 0.60
      lift = 0.6667 / support(Bread=0.7) = 0.9524
5. {Bread, Diapers} (count = 4, support = 0.40)
    - Bread ⇒ Diapers: conf = 4/7 ≈ **0.5714** → **discard**
    - Diapers ⇒ Bread: conf = 4/7 ≈ **0.5714** → **discard**
6. {Butter, Diapers} (count = 4, support = 0.40)
    - Butter ⇒ Diapers: conf = 4/6 = **0.6667** → meets 0.60
      lift = 0.6667 / support(Diapers=0.7) = 0.9524
    - Diapers ⇒ Butter: conf = 4/7 ≈ **0.5714** → **discard**

Rules that meet confidence ≥ 60%

1. **Milk ⇒ Bread**
    - support = 4/10 = **0.40** (40%)
    - confidence = 4/6 = **66.67%**
    - lift = 0.6667 / 0.70 = **0.9524**
2. **Butter ⇒ Bread**
    - support = 4/10 = **0.40**
    - confidence = 4/6 = **66.67%**
    - lift = **0.9524**
3. **Butter ⇒ Diapers**
    - support = 4/10 = **0.40**
    - confidence = 4/6 = **66.67%**
    - lift = **0.9524**

Interpretation and comments

- All three accepted rules have **confidence ≈ 66.7%** and **lift ≈ 0.95** (slightly below 1).
- **Lift < 1** indicates the consequent is actually *less likely* given the antecedent than at baseline — this happens because Bread and Diapers are quite common in the dataset (support = 0.7), so even a high confidence can translate into lift ≤ 1.
- In practice, **high confidence** should be interpreted alongside **lift** (and support). A high-confidence rule with lift ≈ 1 may not be very interesting because the consequent is frequent overall.
- No 3-item rules were frequent at the chosen support threshold; raising or lowering thresholds changes results.

Conclusion

- Frequent 1-itemsets (L1): {Milk (6), Bread (7), Butter (6), Diapers (7)}
- Frequent 2-itemsets (L2): {Milk–Bread (4), Milk–Butter (3), Milk–Diapers (3), Bread–Butter (4), Bread–Diapers (4), Butter–Diapers (4)}
- No frequent 3-itemsets (none reached count ≥ 3)
- **Strong rules (min confidence 60%):**
    1. Milk ⇒ Bread (support 0.40, confidence 0.6667, lift 0.9524)
    2. Butter ⇒ Bread (support 0.40, confidence 0.6667, lift 0.9524)
    3. Butter ⇒ Diapers(support 0.40, confidence 0.6667, lift 0.9524)

Clustering in Data Mining

Clustering is an **unsupervised data mining technique** used to group a set of objects (data points) into clusters such that:

- Objects within a cluster are **similar to each other**, and
- Objects in different clusters are **dissimilar**.

Its primary goal is to identify hidden patterns and segment large datasets into smaller, meaningful groups for business applications like customer segmentation, risk management, anomaly detection, customer behaviour analysis, strategic decision making in business and pattern recognition.

Objectives of Clustering

1. Identify hidden patterns in data.
2. Discover customer segments or behavior groups.
3. Reduce data complexity by summarizing large datasets.
4. Support decision-making, marketing, and risk management.

Use of Clustering in Business Applications

Clustering is used across business domains for **segmentation, prediction, and optimization**.

Business Area	Clustering Application	Example
Marketing	Customer segmentation based on purchasing behavior	Group customers into segments like "high-value", "frequent", "occasional buyers".
Retail	Market basket analysis; store layout design	Identify products often bought together and cluster stores by sales performance.
Banking and Finance	Risk assessment and fraud detection	Cluster customers based on credit risk or transaction patterns.
Healthcare	Patient profiling and disease pattern discovery	Group patients by symptoms, treatment response, or age.
Insurance	Claim analysis and policy grouping	Identify clusters of high-risk or low-risk policyholders.
Telecommunications	Customer churn prediction	Group customers based on usage and retention behavior.

Methods of Clustering

There are several clustering approaches, each based on how the similarity between data points is defined and how clusters are formed.

a) Partitioning Methods

These methods divide data into **k non-overlapping clusters** directly.
Each cluster is represented by a **centroid** (mean point).

Algorithm: *K-Means Clustering*

- Choose number of clusters ***k***.
- Initialize ***k*** cluster centers (randomly).
- Assign each data point to the nearest centroid (using distance measure like Euclidean distance).
- Recompute centroids as the mean of points in each cluster.
- Repeat until centroids stabilize.

Example: A retail company uses K-Means to group 1,000 customers into **3 segments**:

- Cluster 1: Price-sensitive customers
- Cluster 2: Brand-loyal customers
- Cluster 3: Occasional buyers

b) Hierarchical Clustering Methods

Builds a hierarchy (tree structure) of clusters, known as a **dendrogram**.

Example:

In customer data, hierarchical clustering may reveal:

- Level 1: High-value vs. low-value customers
- Level 2: Within high-value → online vs. in-store buyers

Problem

Given the 9 two-dimensional points below, cluster them into **k = 3** clusters using **K-Means**.

ID	**(X, Y)**
P1	(2, 10)
P2	(2, 5)
P3	(8, 4)
P4	(5, 8)
P5	(7, 5)
P6	(6, 4)
P7	(1, 2)
P8	(4, 9)
P9	(3, 3)

Initial centroids (chosen):

C1 = (2, 10) (use P1)

C2 = (5, 8) (use P4)

C3 = (1, 2) (use P7)

Step 1: Initial Setup

We had 9 data points (P1–P9), each defined by (X, Y) coordinates.

We chose **k = 3**, so we needed **3 initial centroids**:

- **C1 = (2, 10)**
- **C2 = (5, 8)**
- **C3 = (1, 2)**

We will calculate **Euclidean distance** of each point from all 3 centroids and assign the point to the **nearest centroid** (i.e., the smallest distance).

The Euclidean distance formula is:

$$d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

Step 2: Iteration 1 — Distance Calculations and Assignments

Point	Calculation (summary)	Nearest Centroid	Assigned Cluster
P1 (2,10)	Distance to C1=0 (same point), C2=3.61, C3=8.06	**C1**	Cluster 1
P2 (2,5)	C1=5.0, C2=4.24, C3=3.0	**C3**	Cluster 3
P3 (8,4)	C1=7.62, C2=4.12, C3=8.06	**C2**	Cluster 2
P4 (5,8)	C1=3.61, C2=0.0, C3=7.28	**C2**	Cluster 2
P5 (7,5)	C1=7.21, C2=3.16, C3=7.07	**C2**	Cluster 2
P6 (6,4)	C1=7.81, C2=4.47, C3=5.10	**C2**	Cluster 2
P7 (1,2)	C1=8.06, C2=9.22, C3=0.0	**C3**	Cluster 3
P8 (4,9)	C1=2.24, C2=1.0, C3=7.21	**C2**	Cluster 2
P9 (3,3)	C1=7.07, C2=5.0, C3=2.24	**C3**	Cluster 3

Thus, after Iteration 1:

- **Cluster 1: {P1}**
- **Cluster 2: {P3, P4, P5, P6, P8}**
- **Cluster 3: {P2, P7, P9}**

Recompute centroids (after Iteration 1)

Compute the mean (average) of X and Y for points in each cluster.

- New C1 = mean of {P1} = (2, 10)
- New C2 = mean of {P3(8,4), P4(5,8), P5(7,5), P6(6,4), P8(4,9)}
    - Xmean = (8+5+7+6+4)/5 = 30/5 = 6.0
    - Ymean = (4+8+5+4+9)/5 = 30/5 = 6.0
      → New C2 = (6.0, 6.0)
- New C3 = mean of {P2(2,5), P7(1,2), P9(3,3)}
    - Xmean = (2+1+3)/3 = 6/3 = 2.0
    - Ymean = (5+2+3)/3 = 10/3 ≈ 3.333333...
      → New C3 = (2.0, 3.333333...)

Iteration 2 — Reassign using new centroids

Centroids now: C1= (2,10), C2 = (6,6), C3=(2,3.3333)

Compute distances and assign:

Point	d to C1	d to C2	d to C3	Assigned cluster
P1 (2,10)	0.00	5.39	6.90	C1
P2 (2,5)	5.00	4.12	1.67	C3
P3 (8,4)	7.62	2.83	6.08	C2
P4 (5,8)	3.61	2.24	5.70	C2
P5 (7,5)	7.21	1.41	5.10	C2
P6 (6,4)	7.81	2.24	4.47	C2
P7 (1,2)	8.06	6.71	1.67	C3
P8 (4,9)	2.24	3.16	5.85	C1
P9 (3,3)	7.07	3.61	0.67	C3

Cluster membership after Iteration 2:

- **Cluster 1 (C1): {P1, P8}**
- **Cluster 2 (C2): {P3, P4, P5, P6}**
- **Cluster 3 (C3): {P2, P7, P9}**

Recompute centroids (after Iteration 2)

- New C1 = mean of {P1(2,10), P8(4,9)}
  - Xmean = (2+4)/2 = 3.0
  - Ymean = (10+9)/2 = 9.5
    → C1 = (3.0, 9.5)
- New C2 = mean of {P3(8,4), P4(5,8), P5(7,5), P6(6,4)}
  - Xmean = (8+5+7+6)/4 = 26/4 = 6.5
  - Ymean = (4+8+5+4)/4 = 21/4 = 5.25
    → C2 = (6.5, 5.25)
- New C3 = mean of {P2(2,5), P7(1,2), P9(3,3)} (unchanged)
  → C3 = (2.0, 3.333333...) = (2, 3.33)

Iteration 3 — Reassign using centroids C1=(3.0,9.5), C2=(6.5,5.25), C3=(2.0,3.3333)

Point	**d to C1**	**d to C2**	**d to C3**	**Assigned cluster**
P1 (2,10)	1.12	5.39	6.90	C1
P2 (2,5)	4.72	4.80	1.67	C3
P3 (8,4)	7.28	1.30	6.08	C2
P4 (5,8)	1.80	2.96	5.70	C1
P5 (7,5)	6.10	1.75	5.10	C2
P6 (6,4)	6.40	1.25	4.47	C2
P7 (1,2)	8.79	6.10	1.67	C3
P8 (4,9)	0.50	3.97	5.85	C1
P9 (3,3)	6.53	3.20	0.67	C3

Cluster membership after Iteration 3:

- **Cluster 1 (C1): {P1, P4, P8}**
- **Cluster 2 (C2): {P3, P5, P6}**
- **Cluster 3 (C3): {P2, P7, P9}**

Recompute centroids (after Iteration 3)

- New C1 = mean of {P1(2,10), P4(5,8), P8(4,9)}
  - Xmean = (2+5+4)/3 = 11/3 = 3.6666667
  - Ymean = (10+8+9)/3 = 27/3 = 9.0
    → C1 = (3.6667, 9.0)
- New C2 = mean of {P3(8,4), P5(7,5), P6(6,4)}
  - Xmean = (8+7+6)/3 = 21/3 = 7.0
  - Ymean = (4+5+4)/3 = 13/3 ≈ 4.3333333
    → C2 = (7.0, 4.3333)
- New C3 = mean of {P2(2,5), P7(1,2), P9(3,3)} (unchanged)
  → C3 = (2.0, 3.3333333)

Iteration 4 — Reassign using centroids C1=(3.6667,9.0), C2=(7.0,4.3333), C3=(2.0,3.3333)

Compute distances (summary):

- After computing distances, each point remains assigned as in Iteration 3:

Cluster membership after Iteration 4:

- Cluster 1: {P1, P4, P8}
- Cluster 2: {P3, P5, P6}
- Cluster 3: {P2, P7, P9}

Because assignments did **not change**, the algorithm has **converged**.

Final result (clusters and centroids)

Cluster A (Centroid ≈ (3.6667, 9.0))
Points: P1 (2,10), P4 (5,8), P8 (4,9)

Cluster B (Centroid ≈ (7.0, 4.3333))
Points: P3 (8,4), P5 (7,5), P6 (6,4)

Cluster C (Centroid ≈ (2.0, 3.3333))
Points: P2 (2,5), P7 (1,2), P9 (3,3)

Interpretation (business/analytical)

- **Cluster A** groups the **high-Y values** (customers with high "Y" metric such as high spending) — good candidates for premium offers.
- **Cluster B** groups **medium X, low Y** (e.g., medium income but moderate purchases).
- **Cluster C** groups **low X and low Y** (low-income/low-spend customers).

K-Means successfully partitioned the dataset into three coherent groups that can be used for targeted strategies (marketing, inventory, personalization).

Problem on Regression

calculate a and b by solving the normal equations of the linear regression curve.

x	y	x^2	xy
8	11	64	88
5	10	25	50
4	4	16	16
6	8	36	48
7	9	49	63
9	13	81	117
10	15	100	150
3	6	9	18
2	12	4	24
12	7	144	84

From the above table

- n=10 , ∑ x = 66 , ∑ y = 95 , ∑ xy =1186 , ∑ x^2 = 528
- Now the normal equations become :
- 95 = 10*b + 66a
- 1186 = 66*b + 528a
- By solving the above two euations we get a = 6.05 and b = -30.429
- The linear regression equation is y = -30.429 + 6.05 x.

CLUSTER ANALYSIS (CLUSTERING)

It is an **unsupervised learning** technique used in data mining to group a set of objects into clusters (groups) such that **objects in the same cluster are more similar to each other** than to those in other clusters. It helps uncover hidden patterns, relationships, or structures in data without predefined labels.

The goal is to ensure that data points within a cluster are more similar to each other than to those in other clusters. For example, in e-commerce retailers use clustering to group customers based on their purchasing habits. If one group frequently buys fitness gear while another prefers electronics. This helps companies to give personalized recommendations and improve customer experience. It is useful for:

1. **Scalability:** It can efficiently handle large volumes of data.
2. **High Dimensionality:** Can handle high-dimensional data.
3. **Adaptability to Different Data Types:** It can work with numerical data like age, salary and categorical data like gender, occupation.
4. **Handling Noisy and Missing Data:** Usually, datasets contain missing values or inconsistencies and clustering can manage them easily.
5. **Interpretability:** Output of clustering is easy to understand and apply in real-world scenarios.

Points to Remember:

One group is treated as a cluster of data objects

- In the process of cluster analysis, the first step is to partition the set of data into groups with the help of data similarity, and then groups are assigned to their respective labels.
- The biggest advantage of clustering over-classification is it can adapt to the changes made and helps single out useful features that differentiate different group

Characteristics of Clustering

1. **Unsupervised Learning:** No predefined output variable — clusters are formed based on data similarity.
2. **Similarity-based Grouping:** Objects are grouped using distance or similarity measures (e.g., Euclidean distance, Manhattan distance).
3. **Intra-cluster similarity (high):** Items in the same cluster are very similar.
4. **Inter-cluster similarity (low):** Items in different clusters are as different as possible.
5. **Scalability:** Should efficiently handle large datasets.
6. **Interpretability:** Resulting clusters should be meaningful and easy to interpret.
7. **Automatic Classification:** Data points are automatically grouped without human intervention.

Business Applications of Clustering

Domain	Use Case
Marketing & Customer Segmentation	Identify customer groups based on purchasing behavior.
Retail & E-commerce	Product recommendation and basket analysis.
Banking & Finance	Credit scoring, fraud detection, and customer profiling.
Healthcare	Group patients with similar symptoms or medical history.
Insurance	Risk assessment and claim pattern analysis.
Telecom	Identifying churn segments and usage patterns.

Uses of cluster analysis:

- It is widely used in many applications such as image processing, data analysis, and pattern recognition.
- It helps marketers to find the distinct groups in their customer base and they can characterize their customer groups by using purchasing patterns.
- It can be used in the field of biology, by deriving animal and plant taxonomies and identifying genes with the same capabilities.
- It also helps in information discovery by classifying documents on the web.

Distance Metrics

Distance metrics are simple mathematical formulas to figure out how similar or different two data points are. Type of distance metrics we choose plays a big role in deciding clustering results. Some of the common metrics are:

- **Euclidean Distance:** It is the most widely used distance metric. It is defined as the distance between two points in Euclidean space. To find the distance between two points, the length of the line segment that connects the two points should be measured.
- It is like **measuring the straightest and shortest path between two points**. This **metric is** widely utilized in various fields such as machine learning, data analysis, computer vision, and many more applications of AI and ML.

Euclidean Distance Formula

Consider two points (x1, y1) and (x2, y2) in a 2-dimensional space; the Euclidean Distance between them is given by using the formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Euclidean Distance in 3D

If the two points (x1, y1, z1) and (x2, y2, z2) are in a 3-dimensional space, the Euclidean Distance between them is given by using the formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

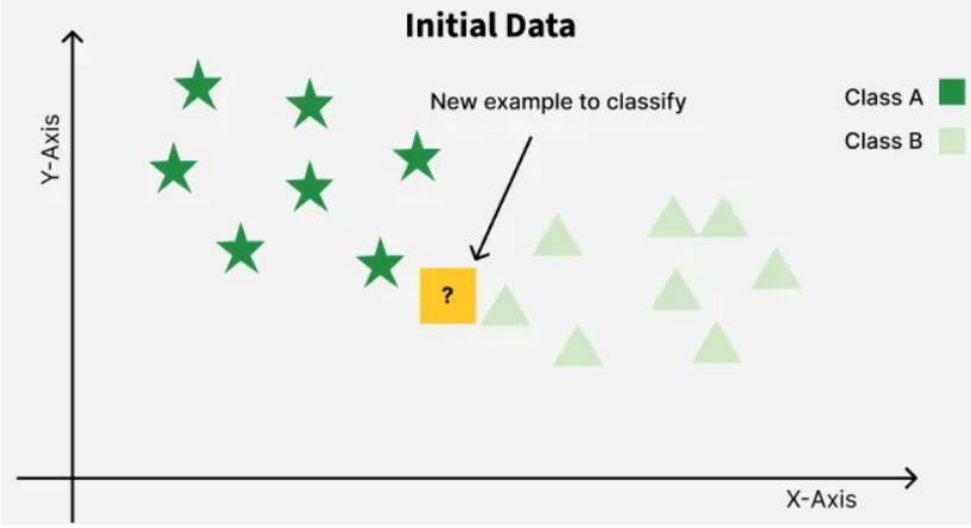
Manhattan Distance: It measures the distance between two points based on grid-like path. It adds the absolute differences between the values.

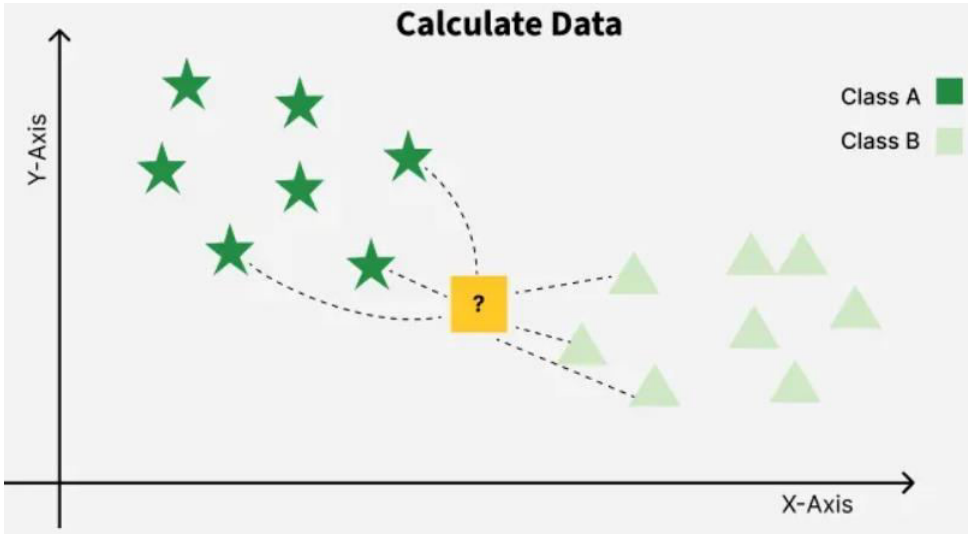
The **Manhattan Distance** between two points **(X1, Y1)** and **(X2, Y2)** is given by **|X1 – X2| + |Y1 – Y2|**. An example is as follows:

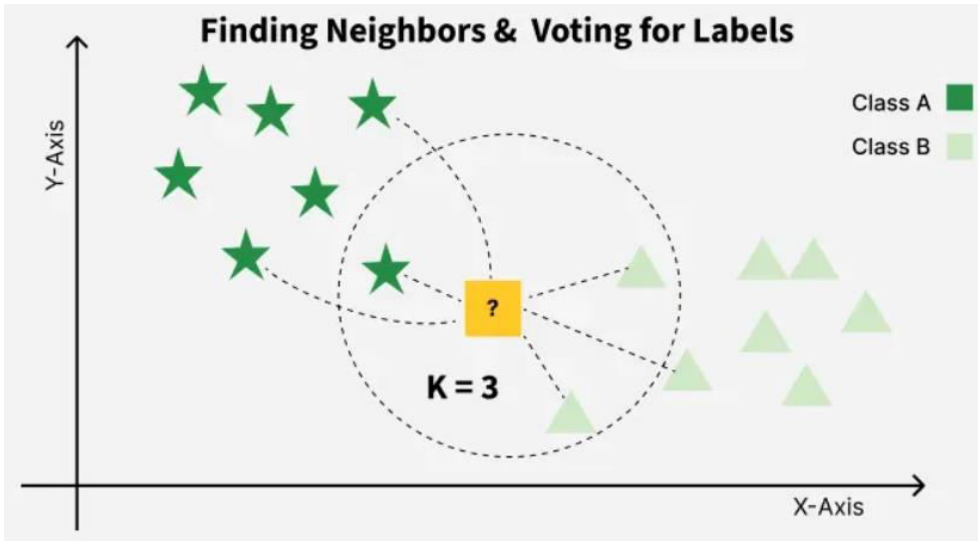
Input: arr[] = {(1, 2), (2, 3), (3, 4)}
Output: 4
Explanation:
The maximum Manhattan distance is found between (1, 2) and (3, 4) i.e.,
|3 - 1| + |4- 2 | = 4.

K-Nearest Neighbour(KNN) Algorithm

KNN is a supervised ML algorithm generally used for classification but can also be used for regression tasks. It works by finding the "k" closest data points (neighbors) to a given input and makes predictions based on the majority class (for classification) or the average value (for regression).



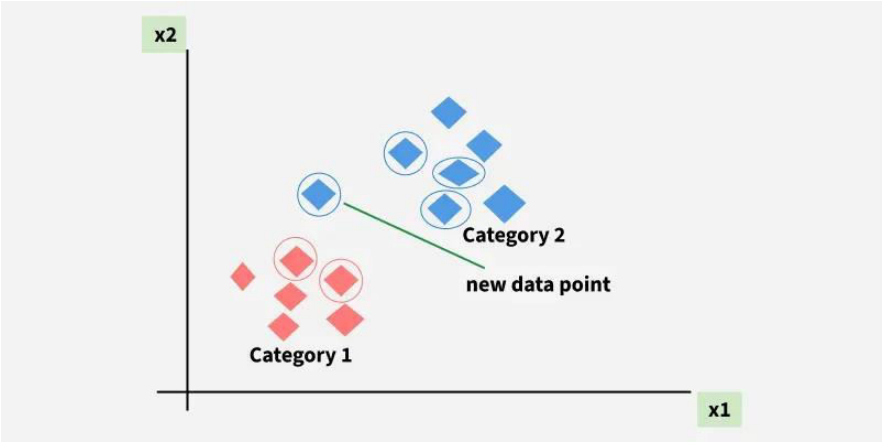




What is the K-Nearest Neighbors Algorithm?

- KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.
- It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data). We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.
- As an example, consider the following table of data points containing two features:

KNN is also called as a lazy learner algorithm because it does not learn from the training set immediately instead it stores the entire dataset and performs computations only at the time of classification. For example, consider the following table of data points containing two features:



The new point is classified as Category 2 because most of its closest neighbours are blue squares. KNN assigns category based on the majority of nearby points.

- The red diamonds represent Category 1 and the blue squares represent Category 2.
- The new data point checks its closest neighbors (circled points).
- Since the majority of its closest neighbors are blue squares (Category 2) KNN predicts the new data point belongs to Category 2.

KNN works by using proximity and majority voting to make predictions.

What is 'K' in K Nearest Neighbour?

In the k-Nearest Neighbours algorithm k is just a number that tells the algorithm how many nearby points or neighbours to look at when it makes a decision.

Example: Imagine you're deciding which fruit it is based on its shape and size. You compare it to fruits you already know.

- If k = 3, the algorithm looks at the 3 closest fruits to the new one.
- If 2 of those 3 fruits are apples and 1 is a banana, the algorithm says the new fruit is an apple because most of its neighbours are apples.

How to choose the value of k for KNN Algorithm?

- The value of k in KNN decides how many neighbours the algorithm looks at when making a prediction. Choosing the right k is important for good results.
- If the data has lots of noise or outliers, using a larger k can make the predictions more stable.
- But if k is too large the model may become too simple and miss important patterns and this is called underfitting.
- So k should be picked carefully based on the data. Rule of thumb: $k \approx \sqrt{n}$

Why KNN Algorithm

- kNN algorithm is a simple, non-parametric, and instance-based machine learning method used for classification and regression. It does not require any assumptions about the underlying data distribution; can also handle both numerical and categorical data.
- Makes predictions based on the similarity of data points in a given dataset. K-NN is less sensitive to outliers compared to other algorithms. It works by finding the K nearest neighbours to a given data point based on a **distance metric**, such as Euclidean distance, which measures the similarity between data points.
- The class or value of the data point is then determined by the majority vote or average of the K neighbours. This approach allows algorithm to adapt to different patterns and make predictions based on the local structure of the data.

Steps in kNN using Euclidean Distance

1. **Compute Euclidean distance** between query point and all points in the dataset.
2. **Sort the distances** in ascending order.
3. **Select the k closest points** (nearest neighbors).
4. **Classify (for classification tasks)** by majority voting among the k neighbors
5. **Predict** by averaging the values of the k nearest neighbors

PROBLEM 1 : We have the following dataset with two features (X_1, X_2) and class labels (0 or 1). Classify a query point (3, 3) using k = 3 and Euclidean distance?

Point (X_1, X_2)	Class
(1, 2)	0
(2, 3)	0
(3, 4)	1
(5, 6)	1

Step 1: Compute Euclidean Distance

Euclidean distance between two points A(X_1, X_2) and B(Y_1, Y_2) is

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Calculate distances from (3,3) to each point in the dataset:

1. Distance to (1,2)

$$d = \sqrt{(3-1)^2 + (3-2)^2} = \sqrt{4+1} = \sqrt{5} \approx 2.24$$

2. Distance to (2,3)

$$d = \sqrt{(3-2)^2 + (3-3)^2} = \sqrt{1+0} = \sqrt{1} = 1.00$$

3. Distance to (3,4)

$$d = \sqrt{(3-3)^2 + (3-4)^2} = \sqrt{0+1} = \sqrt{1} = 1.00$$

4. Distance to (5,6)

$$d = \sqrt{(3-5)^2 + (3-6)^2} = \sqrt{4+9} = \sqrt{13} \approx 3.61$$

Step 2: Sort Distances

Point (X_1, X_2)	Class	Distance from (3,3)
(2,3)	0	1.0
(3,4)	1	1.0
(1,2)	0	2.24
(5,6)	1	3.61

Step 3: Select k Nearest Neighbours

For k=3, the closest 3 points are as follows:

(2,3) → Class 0

(3,4) → Class 1

(1,2) → Class 0

Step 4: Select k Nearest Neighbours

Class 0 : 2 votes

Class 1 : 1 vote

Since **Class 0 has the majority**, we classify (3,3) as **Class 0.**

Answer : The **query point (3, 3) belongs to Class 0.**

EXAMPLE 2 : We'll classify a new 2-D point $x_0 = (4,4)$ using kNN with k = 3 and also with k=5. We will compute distances digit-by-digit to avoid mistakes.

ID	x1	x2	Class
A	1	2	Red
B	2	3	Red
C	3	1	Blue
D	6	5	Blue
E	7	7	Blue
F	8	6	Blue

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Step 1 — compute Euclidean distances to each training point

Distance between (4,4) and A (1,2) = √(9 + 4) = √13 = 3.61

Distance between (4,4) and B (2,3) = √(4 + 1) = √5 = 2.24

Distance between (4,4) and C (3,1) = √(1 + 9) = √10 = 3.16

Distance between (4,4) and D (6,5) = √(4 + 1) = √5 = 2.24

Distance between (4,4) and E (7,7) = √(9 + 9) = √18 = 4.24

Distance between (4,4) and F (8,6) = √(16 + 4) = √20 = 4.48

Step 2 — sort neighbors by distance

Sorted distances (nearest first):

Sorted distances (nearest first):

1. B: 2.24 — Class = Red
2. D: 2.24 — Class = Blue
3. C: 3.16 — Class = Blue
4. A: 3.61 — Class = Red
5. E: 4.24 — Class = Blue
6. F: 4.48 — Class = Blue

(Notice B and D are tied at ≈2.23607; handle ties by including both — fine since k will include both.)

Case 1: k = 3 (majority vote)

Top 3 neighbors: B (Red), D (Blue), C (Blue)

Counts:

- Blue: 2 (D, C)
- Red: 1 (B)

Majority → **Blue**. So $\hat{y} = \text{Blue}.$

Case 2: k = 5 (majority vote)

Top 5 neighbors: B (Red), D (Blue), C (Blue), A (Red), E (Blue)

Counts:

- Blue: 3 (D, C, E)
- Red: 2 (B, A)

Majority → **Blue**. So $\hat{y} = \text{Blue}.$

Problem3 :

We had a dataset of six representing **students' study habits and their exam results**. We need to **classify a new student (S_7)** with this data :

Hours of Study = 3, Classes Attended = 7 using k=3 (i.e., look for 3 nearest neighbors).

Student	Hours of Study (X_1)	Number of Classes Attended (X_2)	Result (Y)
S_1	7	7	Pass
S_2	7	4	Pass
S_3	3	4	Fail
S_4	1	4	Fail
S_5	2	3	Fail
S_6	6	6	Pass

SOLUTION : **Step 1: Compute Euclidean Distances from (3,7)**

Student	Coordinates	Distance from S_7 (3,7)	Result
S_1	(7,7)	√((7−3)² + (7−7)²) = √(16 + 0) = **4.00**	Pass
S_2	(7,4)	√((7−3)² + (4−7)²) = √(16 + 9) = **5.00**	Pass
S_3	(3,4)	√((3−3)² + (4−7)²) = √(0 + 9) = **3.00**	Fail
S_4	(1,4)	√((1−3)² + (4−7)²) = √(4 + 9) = **3.61**	Fail
S_5	(2,3)	√((2−3)² + (3−7)²) = √(1 + 16) = **4.12**	Fail
S_6	(6,6)	√((6−3)² + (6−7)²) = √(9 + 1) = **3.16**	Pass

Step 2: Sort by Distance (Nearest to Farthest)

Rank	Student	Distance	Result
1	S_3	3.00	Fail
2	S_6	3.16	Pass
3	S_4	3.61	Fail
4	S_1	4.00	Pass
5	S_5	4.12	Fail
6	S_2	5.00	Pass

Step 3: Select k = 3 Nearest Neighbours

The **3 nearest neighbours** to S_7 are:

Student	Distance	Result
S_3	3.00	Fail
S_6	3.16	Pass
S_4	3.61	Fail

Step 4: Apply Majority Voting Rule

Among the **3 nearest neighbours**:

- **Fail = 2** (S_3 and S_4)
- **Pass = 1** (S_6)

Therefore, the **predicted result for S_7 = Fail**

Step 5: Interpretation

This means: A student who studies 3 hours and attends 7 classes is likely to **Fail**, based on the patterns learned from historical data.

Advantages of kNN

1. Simple and intuitive (no complex model training).
2. Effective for small datasets.
3. Works well when decision boundaries are irregular.

Limitations

1. Computatssionally expensive for large datasets (distance must be computed for all points).
2. Sensitive to irrelevant or scaled features.
3. Performance depends on the right choice of **k** and **distance metric**.

Pros

- Simple to understand and implement.
- No parametric assumptions about data distribution.
- Naturally handles multi-class problems.
- Flexible: choice of distance and weighting.

Cons

- Slow at prediction time for large datasets (unless optimized).
- Memory intensive (store entire training set).
- Sensitive to irrelevant features and feature scaling.
- Performance deteriorates in high dimensions.

Uses / Applications of KNN Rule in business and industry

- **Customer classification**: label a new customer into segments based on past customers.
- **Credit scoring**: nearest historical borrowers to estimate default risk.
- **Recommendation (cold-start)**: content/item similarity (kNN on item features).
- **Fraud detection**: compare transaction to neighbors to spot anomalies.
- **Medical diagnosis**: classify patient condition from clinical measurements.
- **Image recognition (classical)**: nearest-neighbor on feature descriptors.

Example: Table shows the number of motor registrations in a certain territory for a term of 5 years and the sale of tyres by a firm in that territory

Year	Motor Registrations	No. of Tyres Sold
1	600	1,250
2	630	1,100
3	720	1,300
4	750	1,350
5	800	1,500

Find the regression equation to estimate the sale of tyres when motor registration is known.

Estimate sale of tyres when registration is 850.

Solution: We take registrations as *X* and tyre sales as *Y.* To find line of *Y* on *X*.

X	Y	$d_x = X - \bar{X}$	$d_y = Y - \bar{Y}$	d_x^2	$d_x d_y$
600	*1,250*	*-100*	*-50*	*10,000*	*5,000*
630	*1,100*	*-70*	*-200*	*4,900*	*14,000*
720	*1,300*	*20*	*0*	*400*	*0*
750	*1,350*	*50*	*50*	*2,500*	*2,500*
800	*1,500*	*100*	*200*	*10,000*	*20,000*
$\sum X = 3,500$	$\sum Y = 6,500$	$\sum d_x = 0$	$\sum d_y = 0$	$\sum d_x^2 = 27,800$	$\sum d_x d_y = 41,500$

$$\bar{X} = \frac{\sum X}{N} = \frac{3,500}{5} = 700 \quad \text{and} \quad \bar{Y} = \frac{\sum Y}{N} = \frac{6,500}{5} = 1,300$$

b_{yx} = Regression coefficient of Y on X

$$b_{yx} = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sum (X - \bar{X})^2} = \frac{\sum d_x d_y}{\sum d_x^2} = \frac{4,1500}{2,7800} = 1.4928$$

The regression line of Y on X is given by the equation:

$$Y - \bar{Y} = b_{yx} (X - \bar{X})$$

or $$Y - 1300 = 1.4928 (X - 700)$$

$$Y = 1.4928 X + 255.04$$

When $X = 850$, the value of Y can be calculated from the above equation, by putting $X = 850$

***Y* = (1.4928 *x* 850) + 255. 04 = 1523.92 = 1524 tyres**

Example: For some bivariate data, the following results were obtained

Mean value of variable X = 53.2

Mean value of variable Y = 27.9

Regression coefficient of Y on X = - 1.5

Regression coefficient of X on Y = - 0.2

What is the most likely value of Y, when $X = 60$?

What is the coefficient of correlation between X and Y?

$\overline{X}$ = 53.2 $\overline{Y}$ = 27.9

b_{yx} = -1.5 b_{xy} = -0.2

To obtain value of Y for $X = 60$, we establish the regression line of Y on X,

$$Y - \overline{Y} = b_{yx}\ (X - \overline{X})$$

$$Y - 27.9 = -1.5\ (X - 53.2)$$

or $$Y = -1.5X + 107.7$$

Putting value of $X = 60$, we obtain

$$Y = -1.5 \times 60 + 107.7$$

$$= 17.7$$

Coefficient of correlation between *X, Y* is given by $r^2 = b_{yx} \cdot b_{xy}$

$$r = \pm\sqrt{0.3} = \pm 0.5477$$

Since both regression coefficients are negative, we assign negative value to correlation coefficient r, and so r = -0.5477

Association Rule in Data Mining

Association Rule Mining is a data mining technique used to find relationships or correlations among items in large transactional databases. It helps answer questions like:

"If a customer buys item X, how likely are they to buy item Y?"

This technique is most famously used in **market basket analysis** — e.g., finding that *"70% of customers who buy bread also buy butter."*

Structure of an Association Rule

A rule is generally written as: $X \rightarrow Y$ where **X** and **Y** are itemsets. **X** denotes the antecedent (if-part) and **Y** denotes the consequent (then-part)

Example: Bread $\rightarrow$ Butter means customers who buy Bread are likely to buy Butter.

Association Rule Metrics

To evaluate the strength of association rules, three key measures are used:

(a) Support - Support shows how frequently the rule occurs in the dataset.

$$\text{Support}(X \Rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Total number of transactions}}$$

Support tells how popular (frequency of both items) the rule is in the overall dataset.

(b) Confidence - shows how often items in Y appear in transactions that contain X.

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Transactions containing X}}$$

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

It measures the **reliability** of the inference made by the rule.

(c) Lift - measures how much more likely Y is bought when X is bought.

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

It is the ratio of the observed support to that expected if X and Y were expected (measures the degree of corelation between Xand Y).

If **Lift > 1** → positive correlation between X and Y (items occur together more than expected)

If **Lift = 1** → X and Y are independent.

If **Lift < 1** → negative correlation.

Numerical Problem- A store has the following transaction data:

Transaction ID	Items Purchased
T1	Milk, Bread, Butter
T2	Bread, Butter
T3	Milk, Bread
T4	Milk, Bread, Butter
T5	Bread, Butter

Find the **Support**, **Confidence**, and **Lift** for the rule: Bread → Butter

SOLUTION **Step 1: Total Transactions**
Total = **5**

Step 2: Count occurrences

- Transactions containing **Bread** = T1, T2, T3, T4, T5 → **5**
- Transactions containing **Butter** = T1, T2, T4, T5 → **4**
- Transactions containing **Bread and Butter** = T1, T2, T4, T5 → **4**

Step 3: Compute Metrics

(a) Support of Rule

$$\text{Support}(Bread \Rightarrow Butter) = \frac{4}{5} = 0.8$$

→ **80% of all transactions** contain both Bread and Butter.

(b) Confidence of Rule

$$\text{Confidence}(Bread \Rightarrow Butter) = \frac{4}{5} = 0.8$$

→ When Bread is bought, Butter is also bought **80%** of the time.

(c) Lift of Rule

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

$$\text{Lift}(Bread \Rightarrow Butter) = \frac{0.8}{\text{Support}(\text{Butter})} = \frac{0.8}{0.8} = 1.0$$

Since **Lift = 1**, Bread and Butter are **independent** — buying Bread does not particularly increase or decrease the likelihood of buying Butter.

5. Interpretation

- **High Support (0.8)** → Rule is common.
- **High Confidence (0.8)** → Rule is reliable.
- **Lift = 1.0** → No extra influence; items are independent.

Problem 2: Multi-Item Antecedent Rule

TID	Items Purchased
T1	Milk, Bread, Butter, Eggs
T2	Bread, Butter
T3	Milk, Bread, Butter
T4	Bread, Eggs
T5	Milk, Bread, Eggs

Find Support, Confidence, and Lift for the rule (Milk, Bread) → Butter

Solution

- Total transactions = **5**
- Transactions with (Milk & Bread) = T1, T3, T5 → **3**
- Transactions with (Milk, Bread & Butter) = T1, T3 → **2**
- Transactions with (Butter) = T1, T2, T3 → **3**

$$\text{Support}(X \Rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Total number of transactions}}$$

Support= Transactions with (Milk, Bread & Butter) ÷ Total transactions = 2/5= **0.4(40%)**

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Transactions containing X}}$$

Confidence = 2/3 = 0.667 (67%)

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

Lift = 2/3 ÷ (3/5) = **10/9** = **1.11 which is > 1, implies a positive association.**

Interpretation: When both Milk and Bread are bought, customers are 11% more likely to buy Butter than average → slight positive association.

Problem 3: Complex Case — Multi-item Consequent

TID	Items Purchased
T1	Bread, Butter, Milk, Eggs
T2	Bread, Milk
T3	Bread, Butter
T4	Milk, Eggs
T5	Bread, Butter, Eggs

Find Support, Confidence, and Lift for the rule Milk → (Bread, Butter)

Solution

- Total transactions = 5
- Transactions with Milk = T1, T2, T4 → 3
- Transactions with Bread & Butter = T1, T3, T5 → 3
- Transactions with Milk, Bread & Butter = T1 → 1

Support = 1/5 = 0.2(20%)

Confidence = 1/3 = 0.333

Lift = 1/3 ÷ 3/5 = **5/9 = 0.555** which is < 1

Interpretation: Lift < 1 → Customers who buy Milk are less likely to buy both Bread and Butter together compared to the general population.

Problem 4: Comparing Two Rules

TID	Items Purchased
T1	Pen, Notebook, Eraser
T2	Pen, Notebook
T3	Pen, Eraser
T4	Notebook, Pencil
T5	Pen, Notebook, Pencil

Compare rules: Pen $\rightarrow$ Notebook **and** Notebook $\rightarrow$ Pen.

Solution : The total transactions = 5

Transactions with Pen = T1, T2, T3, T5 → 4

Transactions with Notebook = T1, T2, T4, T5 → 4

Transactions with both = T1, T2, T5 → 3

For Rule 1 : Pen $\rightarrow$ Notebook

Support = 3/5 = 0.6
Confidence = 3/4 = 0.75
Lift = 0.75 / 0.8 = 0.9375

For Rule 2 : Notebook $\rightarrow$ Pen

Support = 0.6
Confidence = 3/4 = 0.75
Lift = 0.75 / 0.8 = 0.9375

Interpretation: Both rules are symmetric, and Lift < 1 → **slightly negative correlation**

Apriori Algorithm in Data Mining

It is a basic method used in DM to find groups of items that often appear together in large sets of data. It helps to discover useful patterns or rules about how items are related which is particularly valuable in market basket analysis. *Like in a grocery store if many customers buy bread and butter together, the store can use this information to place these items closer or create special offers. This helps the store sell more and make customers happy*

This **algorithm** is one of the **most fundamental algorithms** in data mining, specifically used for **association rule learning** — to find frequent itemsets and derive meaningful **association rules** (like *"Customers who buy bread also buy butter"*).

It is based on the principle that:

"All subsets of a frequent itemset must also be frequent."

This principle is called the **Apriori property.**

Objective: To find

- All **frequent itemsets** that occur above a given **minimum support threshold**.
- Strong **association rules** that have **confidence** above a given **minimum confidence threshold**.

Key Concepts and Formulas

Term	Definition	Formula
Support (A ⇒ B)	Frequency of transactions containing both A and B	$\text{Support}(A \Rightarrow B) = \frac{n(A \cup B)}{N}$
Confidence (A ⇒ B)	Likelihood of buying B when A is bought	$\text{Confidence}(A \Rightarrow B) = \frac{n(A \cup B)}{n(A)}$
Lift (A ⇒ B)	Strength of rule compared to random co-occurrence	$\text{Lift}(A \Rightarrow B) = \frac{\text{Confidence}(A \Rightarrow B)}{\text{Support}(B)}$

Steps of the Apriori Algorithm

1. **Set minimum support (min_sup)** and **minimum confidence (min_conf)** thresholds.
2. **Generate all 1-itemsets** and count their frequency.
3. **Prune** itemsets that do not meet min_sup.
4. Use frequent itemsets of size k to generate **candidate itemsets of size (k+1)** (self-join).
5. **Count frequency** of these candidates and again prune those not meeting min_sup.
6. Continue until no more frequent itemsets can be generated.
7. From the frequent itemsets, **generate strong association rules** using the min_conf threshold.

Numerical Example

TID	Items Purchased
T1	Milk, Bread, Butter
T2	Bread, Butter
T3	Milk, Bread
T4	Milk, Bread, Butter
T5	Bread, Butter, Eggs

Given: Min. Support = 60% (i.e., 0.6), Min. Confidence = 70% (i.e., 0.7) and transactions = 5

Step 1 : Generate 1-itemsets

Item	Count	Support
Milk	3	3/5 = 0.6
Bread	5	5/5 = 1.0
Butter	4	4/5 = 0.8
Eggs	1	1/5 = 0.2

Step 2: Generate 2-itemsets

Possible pairs: (Milk, Bread), (Milk, Butter), (Bread, Butter)

Itemset	Count	Support
(Milk, Bread)	3	0.6
(Milk, Butter)	2	0.4
(Bread, Butter)	4	0.8

Frequent 2-itemsets: (Milk, Bread), (Bread, Butter) as they meet minimum support criteria

Step 3: Generate 3-itemsets

Possible 3-itemset: (Milk, Bread, Butter)

Itemset	Count	Support
(Milk, Bread, Butter)	2	0.4

Not frequent (Support = 0.4 < 0.6)

Final frequent itemsets: {Milk}, {Bread}, {Butter}, {Milk, Bread}, {Bread, Butter}

Step 4: Generate Association Rules

From **2-itemsets**, we can derive:

Rule 1: Milk ⇒ Bread

Support = 0.6

Confidence = (3/3) = 1.0 ✅

Lift = 1.0 / 1.0 = 1.0

Rule 2: Bread ⇒ Milk

Support = 0.6

Confidence = (3/5) = 0.6 ❌ (below 0.7 threshold)

Rule 3: Bread ⇒ Butter

Support = 0.8

Confidence = (4/5) = 0.8 ✅

Lift = 0.8 / 0.8 = 1.0

Rule 4: Butter ⇒ Bread

Support = 0.8

Confidence = (4/4) = 1.0 ✅

Lift = 1.0 / 1.0 = 1.0

Strong Rules (min_conf ≥ 0.7):

Milk ⇒ Bread (Confidence = 1.0)

Bread ⇒ Butter (Confidence = 0.8)

Butter ⇒ Bread (Confidence = 1.0)

Interpretation

- Milk ⇒ Bread → Every customer who buys Milk also buys Bread.
- Butter ⇒ Bread → Butter sales are always accompanied by Bread purchases.
- Bread ⇒ Butter → Most Bread buyers also buy Butter (80% of the time).

These insights help in cross-selling and product placement strategies.

Summary table

Rule	Support	Confidence	Lift	Strength
Milk ⇒ Bread	0.6	1.0	1.0	Strong
Bread ⇒ Butter	0.8	0.8	1.0	Strong
Butter ⇒ Bread	0.8	1.0	1.0	Strong

- **Apriori** uses the “subset” property to reduce computation.
- It is **bottom-up** — starting with 1-itemsets and building up.
- Performance depends on **support/confidence thresholds**.

REGRESSION ANALYSIS IN DATA MINING

Regression Analysis is a **predictive modeling technique** used to understand the relationship between a **dependent variable (target)** and one or more **independent variables (predictors)**. In **data mining**, regression is used to **forecast trends**, **predict numeric outcomes**, and **analyze relationships** between variables — for example:

- Predicting sales revenue based on advertising expenditure.
- Estimating house prices based on size, location, and number of rooms.
- Forecasting stock prices based on past data.

Purpose of Regression Analysis in Data Mining

Regression is part of **predictive analytics**, and it helps in:

- **Prediction:** Estimating future or unknown values.
- **Trend Analysis:** Understanding how variables change over time.
- **Relationship Identification:** Measuring how strongly variables are related.
- **Optimization:** Finding values that maximize or minimize outcomes.

Types of Regression Used in Data Mining

Type	Description	Example
Linear Regression	Relationship between variables is linear (straight-line).	Predicting sales based on advertising spend.
Multiple Linear Regression	More than one independent variable.	Predicting house prices based on area, rooms, location.
Logistic Regression	Used when the dependent variable is categorical (e.g., Yes/No).	Predicting if a customer will buy a product.
Polynomial Regression	Non-linear relationships fitted by higher-order equations.	Predicting growth rates or curved trends.

Linear Regression Model

The simplest regression is the **Simple Linear Regression**, represented as: $Y=a + bX + \varepsilon$

Where Y is the dependent variable (Predicted variable), X is the independent variable (Predictor), a is the Intercept (value of Y when X = 0), b is the regression coefficient or slope of the line and ε denotes Error term (unexplained variation) which is assumed as zero. So, the regression line of Y on X is written as $Y=a + bX$, where a, b are to be determined using the dataset values of (X,Y).

Numerical Problem:

A marketing manager wants to understand how **advertising expenditure (X)** affects **sales revenue (Y)**. The data collected from 5 months is:

Month	Advertising (X in ₹'000)	Sales (Y in ₹'000)
1	2	4
2	3	5
3	5	7
4	7	10
5	9	15

Find the **regression equation** and **predict sales** when advertising = ₹6,000.

SOLUTION : Step 1: Compute Required Values

X	Y	X^2	XY
2	4	4	8
3	5	9	15
5	7	25	35
7	10	49	70
9	15	81	135

$$\sum X = 26, \quad \sum Y = 41, \quad \sum X^2 = 168, \quad \sum XY = 263, \quad n = 5$$

Step 2: Find the Slope (b)

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2}$$

$$b = \frac{5(263) - (26)(41)}{5(168) - (26)^2}$$

$$b = \frac{1315 - 1066}{840 - 676} = \frac{249}{164} = 1.518$$

Step 3: Find the Intercept (a)

$$a = \frac{\sum Y - b(\sum X)}{n}$$

$$a = \frac{41 - 1.518(26)}{5} = \frac{41 - 39.468}{5} = 0.306$$

Step 4: Regression Equation

$$Y = a + bX = 0.306 + 1.518X$$

Step 5: Prediction

For $X = 6$ (₹6,000 advertising spend):

$$Y = 0.306 + 1.518(6) = 0.306 + 9.108 = 9.414$$

Predicted Sales = ₹9.41 thousand = ₹9410

Interpretation

1. **Slope (b = 1.518):** For every ₹1,000 increase in advertising, sales increase by ₹1.518 thousand.
2. **Intercept (a = 0.306):** When advertising is zero, sales are expected to be ₹0.306 thousand.
3. Thus, there is a **strong positive linear relationship** between advertising and sales.

Uses of Regression Analysis in Data Mining

Area	Application
Marketing	Predicting sales, demand forecasting
Finance	Credit scoring, stock price prediction
Operations	Resource optimization, capacity planning
HR Analytics	Salary prediction, performance analysis
Healthcare	Predicting patient outcomes or treatment costs

CONCLUSION:

- Regression is a **supervised learning technique** used for **numeric prediction**.
- The **strength** of prediction depends on correlation between variables.
- It is a foundational method for **predictive analytics**, **trend forecasting**, and **decision support systems**.

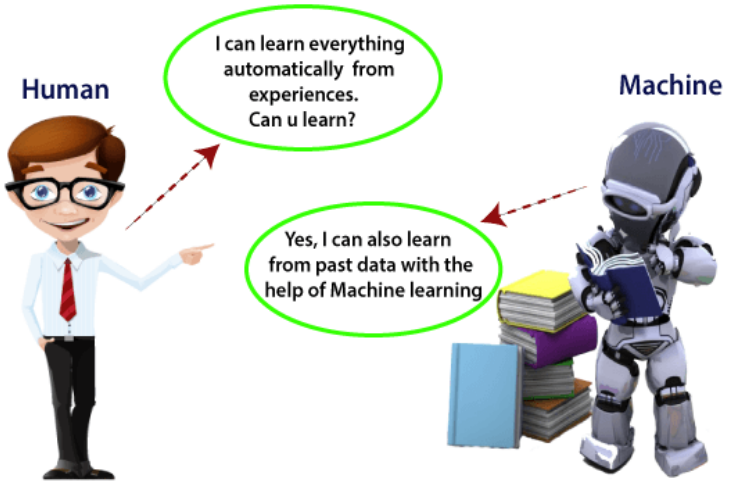
PROBLEM 2 ON REGRESSION

x	y	x^2	xy
8	11	64	88
5	10	25	50
4	4	16	16
6	8	36	48
7	9	49	63
9	13	81	117
10	15	100	150
3	6	9	18
2	12	4	24
12	7	144	84

From the above table

- n=10 , ∑ x = 66 , ∑ y = 95 , ∑ xy =1186 , ∑ x^2 = 528
- Now the normal equations become :
- 95 = 10*b + 66a
- 1186 = 66*b + 528a
- By solving the above two euations we get a = 6.05 and b = -30.429
- The linear regression equation is y = -30.429 + 6.05 x.

Machine Learning - Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for **building mathematical models and making predictions using historical data or information**. Currently, it is being used for various tasks such as **image recognition**, **speech recognition**, **email filtering**, **Facebook auto-tagging**, **recommender system**, and many more.



Machine Learning is said as a subset of AI that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. Arthur Samuel, an early American leader in the field of computer gaming and artificial intelligence, coined the term "Machine Learning " in 1959 while at IBM. He defined machine learning as "**the field of study that gives computers the ability to learn without being explicitly programmed** ". However, there is no universally accepted definition for machine learning

Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

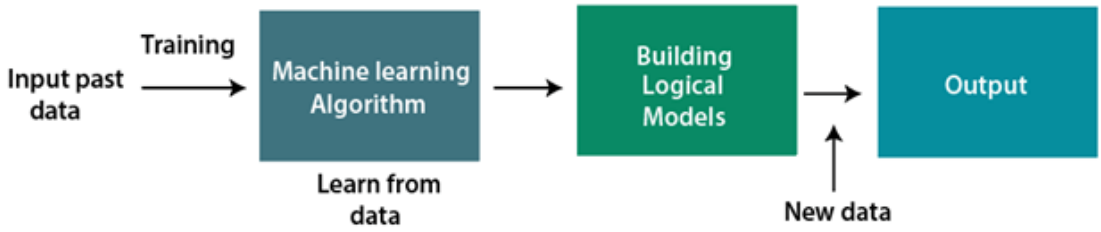
There are several types of machine learning, including **supervised learning, unsupervised learning, and reinforcement learning.**

How does Machine Learning work

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem.

The below block diagram explains the working of Machine Learning algorithm:



Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- ML is much similar to data mining as it also deals with the huge amount of the data.

Need for Machine Learning

The need for ML is increasing day by day. The reason behind the need for ML is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

We can train ML algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of ML can be easily understood by its use-cases. Currently, ML is used in self-driving cars, cyber fraud detection, face recognition, and friend suggestion by Facebook, etc. Various top companies such as Netflix and Amazon have built ML models that are using a vast amount of data to analyze the user interest and recommend product accordingly.

Following are some key points which show the importance of ML :

- Rapid increment in the production of data
- Solving complex problems, which are difficult for a human
- Decision making in various sector including finance
- Finding hidden patterns and extracting useful information from data.

Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. **Supervised learning**
2. **Unsupervised learning**
3. **Reinforcement learning**

Supervised Learning

Supervised learning is a type of ML method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering. Supervised learning can be grouped further in two categories of algorithms:

- **Classification**
- **Regression**

Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. There are two categories of USL algorithms:

- **Clustering**
- **Association**

Reinforcement Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action.

The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

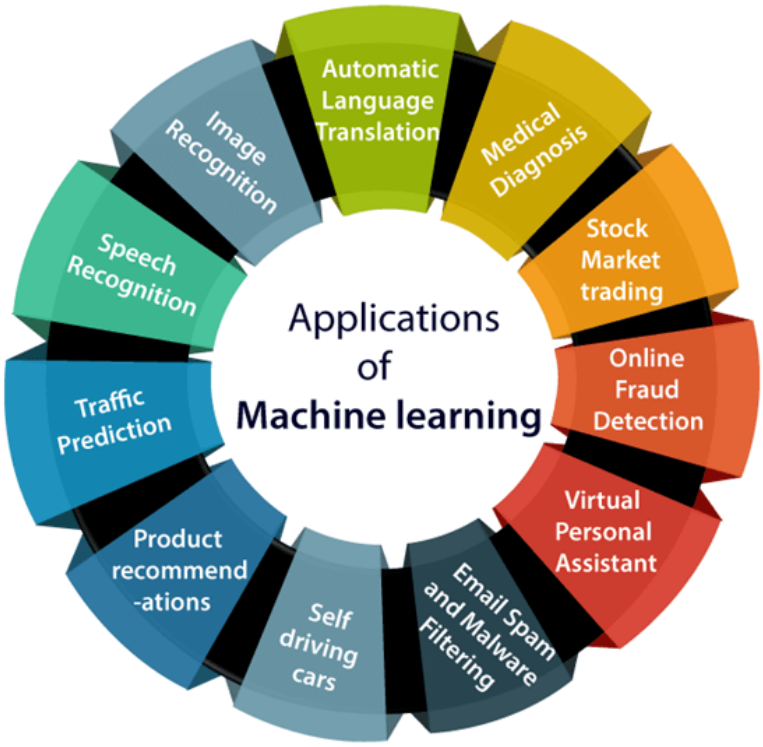
Machine Learning at present:

Now ML has got a great advancement in its research, and it is present everywhere around us, such as self-driving cars, Amazon Alexa, Catboats, recommender system, and many more.

It includes Supervised, unsupervised and reinforcement learning with,
clustering, classification, decision tree, SVM algorithms, etc. Modern machine learning models can be used for making various predictions including **weather prediction, disease prediction, stock market analysis, etc.**

Applications of Machine learning

ML is a buzzword for today's technology, and it is growing very rapidly day by day. We are using ML in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Below are some most trending real-world applications of Machine Learning:



1. Image Recognition:

Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, **Automatic friend tagging suggestion**:

Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's **face detection** and **recognition algorithm**.

It is based on the Facebook project named "**Deep Face**," which is responsible for face recognition and person identification in the picture.

2. Speech Recognition

While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning.

Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**." At present, machine learning algorithms are widely used by various applications of speech recognition. **Google assistant**, **Siri**, **Cortana**, and **Alexa** are using speech recognition technology to follow the voice instructions.

3. Traffic prediction:

If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.

It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:

- **Real Time location** of the vehicle form Google Map app and sensors
- **Average time has taken** on past days at the same time.

Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

4. Product recommendations:

Machine learning is widely used by various e-commerce and entertainment companies such as **Amazon**, **Netflix**, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.

Google understands the user interest using various ML algorithms and suggests the product as per customer interest.As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

5. Self-driving cars:

One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

6. Email Spam and Malware Filtering:

Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Below are some spam filters used by Gmail:

- Content Filter
- Header filter
- General blacklists filter
- Rules-based filters
- Permission filters

Some machine learning algorithms such as **Multi-Layer Perceptron**, **Decision tree**, and **Naïve Bayes classifier** are used for email spam filtering and malware detection.

7. Virtual Personal Assistant:

We have various virtual personal assistants such as **Google assistant**, **Alexa**, **Cortana**, **Siri**. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.

These virtual assistants use machine learning algorithms as an important part.

These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

8. Online Fraud Detection:

Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as **fake accounts**, **fake ids**, and **steal money** in the middle of a transaction. So to detect this, **Feed Forward Neural network** helps us by checking whether it is a genuine transaction or a fraud transaction.

For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.

9. Stock Market trading:

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning's **long short term memory neural network** is used for the prediction of stock market trends.

10. Medical Diagnosis:

In medical science, ML is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain. It helps in finding brain tumors and other brain-related diseases easily.

11. Automatic Language Translation:

Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.

The technology behind the automatic translation is a sequence-to-sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.

UNSUPERVISED ML AND ITS TYPES

1. Clustering: Clustering is the process of grouping similar data points together based on their characteristics or attributes. Clustering algorithms attempt to find patterns in the data and group the data points together based on those patterns.
2. Anomaly detection: Anomaly detection is the process of identifying data points that deviate from the expected behavior of the data set. Anomaly detection algorithms attempt to identify outliers or anomalies in the data.
3. Association rule learning: Association rule learning is the process of discovering relationships between variables in a data set. Association rule learning algorithms attempt to find patterns in the data that indicate that certain variables are associated with each other.
4. Dimensionality reduction: Dimensionality reduction is the process of reducing the number of variables or features in a data set. Dimensionality reduction algorithms attempt to identify the most important features in the data set and remove less important ones.
5. Neural Networks: In unsupervised learning, neural networks can be used for unsupervised representation learning or self-supervised learning. This allows the network to learn useful features of the data without being explicitly given a specific target.

CLUSTERING AND ITS TYPES

K-Means Clustering: It is a popular clustering algorithm that partitions the data into K clusters, where K is a user-defined number. The algorithm works by iteratively assigning data points to their nearest centroid and updating the centroid's position until convergence.

1. Hierarchical Clustering: It is a clustering algorithm that builds a hierarchy of clusters by either merging small clusters into larger ones or splitting large clusters into smaller ones. There are two types of hierarchical clustering: agglomerative (bottom-up) and divisive (top-down).
2. DBSCAN (Density-Based Spatial Clustering of Applications with Noise): It is a clustering algorithm that groups together points that are closely packed together, while marking outliers that lie alone in low-density regions. It works by defining a neighborhood around each data point and grouping points that are densely connected.
3. Mean Shift: It is a non-parametric clustering algorithm that doesn't require specifying the number of clusters beforehand. The algorithm works by iteratively shifting a window (kernel) towards the maximum density of data points until convergence.

4. Spectral Clustering: It is a clustering algorithm that uses the eigenvalues and eigenvectors of a similarity matrix to project the data onto a lower-dimensional space where it can be clustered. The algorithm works by partitioning the graph into two or more clusters based on the eigenvectors.
5. Fuzzy C-Means: It is a soft clustering algorithm that assigns each data point to multiple clusters with varying degrees of membership. The algorithm works by minimizing the sum of the squared distances between the data points and their corresponding cluster centers, weighted by the membership degrees.

REINFORCEMENT LEARNING

Reinforcement learning (RL) is a type of machine learning algorithm that is used for training an agent to make decisions in an environment by interacting with it over time. The agent learns by receiving feedback in the form of rewards or penalties based on its actions, and the goal is to maximize the cumulative reward received over time. Here are the key components of a typical reinforcement learning problem:

Environment: This is the external system or process with which the agent interacts. It can be anything from a simple game environment to a complex real-world scenario such as robotics, traffic control, or finance.

Agent: This is the learner or decision-maker that interacts with the environment. The agent takes actions based on the current state of the environment and receives feedback in the form of rewards or penalties based on the outcomes of its actions.

State: This is a representation of the environment at a given point in time, which the agent uses to make decisions.

Action: This is the decision made by the agent based on the current state of the environment.

Reward: This is the feedback provided to the agent after it takes an action. The reward can be positive or negative and is used to reinforce or discourage certain behaviors.

The reinforcement learning algorithm works by having the agent interact with the environment and learn from its experiences over time. The agent starts by exploring the environment and trying different actions, and the rewards it receives are used to update its policy for taking future actions.

Reinforcement learning is used in a variety of applications, including robotics, game playing, and autonomous vehicles. It is a powerful technique that can learn complex behaviors and can be combined with other machine learning approaches such as deep learning to achieve even better results.

PROBLEM 3: A company's marketing department collected the following data on **advertising expenditure (X)** and **sales revenue (Y)** (both in ₹'000). They want to determine the **regression equation** and **predict sales when advertising = ₹25,000**.

Observation	X (Advertising)	Y (Sales)
1	5	10
2	10	15
3	12	19
4	15	23
5	18	26
6	20	30
7	22	32
8	25	35
9	30	40
10	35	43

Solution:

X	Y	X^2	XY
5	10	25	50
10	15	100	150
12	19	144	228
15	23	225	345
18	26	324	468
20	30	400	600
22	32	484	704
25	35	625	875
30	40	900	1200
35	43	1225	1505

$$\sum X = 212, \quad \sum Y = 273, \quad \sum X^2 = 4452, \quad \sum XY = 6125, \quad n = 10$$

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2}$$

$$b = \frac{10(6125) - (212)(273)}{10(4452) - (212)^2}$$

$$b = \frac{61250 - 57876}{44520 - 44944} = \frac{3374}{576} = 5.86$$

$$a = \frac{\sum Y - b(\sum X)}{n}$$

$$a = \frac{273 - 5.86(212)}{10} = \frac{273 - 1241.32}{10} = \frac{-968.32}{10} = -96.832$$

$$Y = -96.832 + 5.86X$$

Predict sales when **Advertising (X) = 25**: Y = −96.832 + 5.86(25) = 49.668

Problem statement (market-basket)

T1: {Milk, Bread, Butter}

T2: {Bread, Butter, Diapers}

T3: {Milk, Bread, Diapers}

T4: {Milk, Butter, Diapers}

T5: {Bread, Diapers}

T6: {Milk, Bread, Butter}

T7: {Butter, Diapers}

T8: {Milk, Bread}

T9: {Bread, Butter, Diapers}

T10:{Milk, Diapers}

- Minimum support = **30%** (for 10 transactions → min count = 3)
- Minimum confidence = **60%**

Goal: find frequent itemsets and generate association rules meeting min confidence.

SOLUTION TO PROBLEM

Step 1 — Find frequent 1-itemsets (L1)

Count occurrences (support counts and support fraction):

- Milk: appears in T1, T3, T4, T6, T8, T10 → count = **6**, support = 6/10 = **0.60**
- Bread: T1, T2, T3, T5, T6, T8, T9 → count = **7**, support = **0.70**
- Butter: T1, T2, T4, T6, T7, T9 → count = **6**, support = **0.60**
- Diapers: T2, T3, T4, T5, T7, T9, T10 → count = **7**, support = **0.70**

All four items have support ≥ 0.30.

So **L1 = {Milk, Bread, Butter, Diapers}**.

Step 2 — Generate candidate 2-itemsets (C2) and count supports

Candidates (all pairs from L1):

1. {Milk, Bread} — occurs in T1, T3, T6, T8 → count = **4**, support = **0.40**
2. {Milk, Butter} — T1, T4, T6 → count = **3**, support = **0.30**
3. {Milk, Diapers} — T3, T4, T10 → count = **3**, support = **0.30**
4. {Bread, Butter} — T1, T2, T6, T9 → count = **4**, support = **0.40**
5. {Bread, Diapers} — T2, T3, T5, T9 → count = **4**, support = **0.40**
6. {Butter, Diapers} — T2, T4, T7, T9 → count = **4**, support = **0.40**

All candidate pairs meet min support count (≥3).

So **L2 = {MB, MBu, MDi, BBu, BDi, BuDi}** where shorthand: M=Milk, B=Bread, Bu=Butter, Di=Diapers.

Step 3 — Generate candidate 3-itemsets (C3) from L2 and count supports

Possible 3-item combos (from L2 joins):

- {Milk, Bread, Butter} — present in T1, T6 → count = **2**, support = **0.20**
- {Milk, Bread, Diapers} — present in T3 → count = **1**, support = **0.10**
- {Milk, Butter, Diapers} — present in T4 → count = **1**, support = **0.10**
- {Bread, Butter, Diapers} — present in T2, T9 → count = **2**, support = **0.20**

None of these 3-itemsets reach min support count (3). So stop here.

Final frequent itemsets = L1 and L2 (no L3).

Step 4 — Generate association rules from frequent itemsets

We generate candidate rules from frequent 2-itemsets (and could also from larger itemsets, but there are none). For each pair {X,Y} we can form two rules: X ⇒ Y and Y ⇒ X. Use the formulae:

- support(X⇒Y) = count(X ∪ Y) / N
- confidence(X⇒Y) = count(X ∪ Y) / count(X)
- lift(X⇒Y) = confidence(X⇒Y) / support(Y)

We will keep rules with **confidence ≥ 0.60**.

Compute for each frequent pair (counts from above; N = 10):

1. {Milk, Bread} (count = 4, support = 0.40)
    - **Milk ⇒ Bread**
      confidence = 4 / count(Milk=6) = 4/6 = **0.6667** (66.67%) → meets 0.60
      lift = 0.6667 / support(Bread=0.7) = 0.6667 / 0.7 = **0.9524**
    - **Bread ⇒ Milk**
      confidence = 4 / 7 = **0.5714** → below 0.60 (discard)
2. {Milk, Butter} (count = 3, support = 0.30)
    - Milk ⇒ Butter: conf = 3/6 = **0.5** → **discard**
    - Butter ⇒ Milk: conf = 3/6 = **0.5** → **discard**
3. {Milk, Diapers} (count = 3, support = 0.30)
    - Milk ⇒ Diapers: conf = 3/6 = **0.5** → **discard**
    - Diapers ⇒ Milk: conf = 3/7 ≈ **0.4286** → **discard**
4. {Bread, Butter} (count = 4, support = 0.40)
    - Bread ⇒ Butter: conf = 4/7 ≈ **0.5714** → **discard**
    - Butter ⇒ Bread: conf = 4/6 = **0.6667** → meets 0.60
      lift = 0.6667 / support(Bread=0.7) = 0.9524
5. {Bread, Diapers} (count = 4, support = 0.40)
    - Bread ⇒ Diapers: conf = 4/7 ≈ **0.5714** → **discard**
    - Diapers ⇒ Bread: conf = 4/7 ≈ **0.5714** → **discard**
6. {Butter, Diapers} (count = 4, support = 0.40)
    - Butter ⇒ Diapers: conf = 4/6 = **0.6667** → meets 0.60
      lift = 0.6667 / support(Diapers=0.7) = 0.9524
    - Diapers ⇒ Butter: conf = 4/7 ≈ **0.5714** → **discard**

Rules that meet confidence ≥ 60%

1. **Milk ⇒ Bread**
    - support = 4/10 = **0.40** (40%)
    - confidence = 4/6 = **66.67%**
    - lift = 0.6667 / 0.70 = **0.9524**
2. **Butter ⇒ Bread**
    - support = 4/10 = **0.40**
    - confidence = 4/6 = **66.67%**
    - lift = **0.9524**
3. **Butter ⇒ Diapers**
    - support = 4/10 = **0.40**
    - confidence = 4/6 = **66.67%**
    - lift = **0.9524**

Interpretation and comments

- All three accepted rules have **confidence ≈ 66.7%** and **lift ≈ 0.95** (slightly below 1).
- **Lift < 1** indicates the consequent is actually *less likely* given the antecedent than at baseline — this happens because Bread and Diapers are quite common in the dataset (support = 0.7), so even a high confidence can translate into lift ≤ 1.
- In practice, **high confidence** should be interpreted alongside **lift** (and support). A high-confidence rule with lift ≈ 1 may not be very interesting because the consequent is frequent overall.
- No 3-item rules were frequent at the chosen support threshold; raising or lowering thresholds changes results.

Conclusion

- Frequent 1-itemsets (L1): {Milk (6), Bread (7), Butter (6), Diapers (7)}
- Frequent 2-itemsets (L2): {Milk–Bread (4), Milk–Butter (3), Milk–Diapers (3), Bread–Butter (4), Bread–Diapers (4), Butter–Diapers (4)}
- No frequent 3-itemsets (none reached count ≥ 3)
- **Strong rules (min confidence 60%):**
    1. Milk ⇒ Bread (support 0.40, confidence 0.6667, lift 0.9524)
    2. Butter ⇒ Bread (support 0.40, confidence 0.6667, lift 0.9524)
    3. Butter ⇒ Diapers(support 0.40, confidence 0.6667, lift 0.9524)

Clustering in Data Mining

Clustering is an **unsupervised data mining technique** used to group a set of objects (data points) into clusters such that:

- Objects within a cluster are **similar to each other**, and
- Objects in different clusters are **dissimilar**.

Its primary goal is to identify hidden patterns and segment large datasets into smaller, meaningful groups for business applications like customer segmentation, risk management, anomaly detection, customer behaviour analysis, strategic decision making in business and pattern recognition.

Objectives of Clustering

1. Identify hidden patterns in data.
2. Discover customer segments or behavior groups.
3. Reduce data complexity by summarizing large datasets.
4. Support decision-making, marketing, and risk management.

Use of Clustering in Business Applications

Clustering is used across business domains for **segmentation, prediction, and optimization**.

Business Area	Clustering Application	Example
Marketing	Customer segmentation based on purchasing behavior	Group customers into segments like "high-value", "frequent", "occasional buyers".
Retail	Market basket analysis; store layout design	Identify products often bought together and cluster stores by sales performance.
Banking and Finance	Risk assessment and fraud detection	Cluster customers based on credit risk or transaction patterns.
Healthcare	Patient profiling and disease pattern discovery	Group patients by symptoms, treatment response, or age.
Insurance	Claim analysis and policy grouping	Identify clusters of high-risk or low-risk policyholders.
Telecommunications	Customer churn prediction	Group customers based on usage and retention behavior.

Methods of Clustering

There are several clustering approaches, each based on how the similarity between data points is defined and how clusters are formed.

a) Partitioning Methods

These methods divide data into **k non-overlapping clusters** directly.
Each cluster is represented by a **centroid** (mean point).

Algorithm: *K-Means Clustering*

- Choose number of clusters ***k***.
- Initialize ***k*** cluster centers (randomly).
- Assign each data point to the nearest centroid (using distance measure like Euclidean distance).
- Recompute centroids as the mean of points in each cluster.
- Repeat until centroids stabilize.

Example: A retail company uses K-Means to group 1,000 customers into **3 segments**:

- Cluster 1: Price-sensitive customers
- Cluster 2: Brand-loyal customers
- Cluster 3: Occasional buyers

b) Hierarchical Clustering Methods

Builds a hierarchy (tree structure) of clusters, known as a **dendrogram**.

Example:

In customer data, hierarchical clustering may reveal:

- Level 1: High-value vs. low-value customers
- Level 2: Within high-value → online vs. in-store buyers

Problem

Given the 9 two-dimensional points below, cluster them into **k = 3** clusters using **K-Means**.

ID	**(X, Y)**
P1	(2, 10)
P2	(2, 5)
P3	(8, 4)
P4	(5, 8)
P5	(7, 5)
P6	(6, 4)
P7	(1, 2)
P8	(4, 9)
P9	(3, 3)

Initial centroids (chosen):

C1 = (2, 10) (use P1)

C2 = (5, 8) (use P4)

C3 = (1, 2) (use P7)

Step 1: Initial Setup

We had 9 data points (P1–P9), each defined by (X, Y) coordinates.

We chose **k = 3**, so we needed **3 initial centroids**:

- **C1 = (2, 10)**
- **C2 = (5, 8)**
- **C3 = (1, 2)**

We will calculate **Euclidean distance** of each point from all 3 centroids and assign the point to the **nearest centroid** (i.e., the smallest distance).

The Euclidean distance formula is:

$$d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

Step 2: Iteration 1 — Distance Calculations and Assignments

Point	Calculation (summary)	Nearest Centroid	Assigned Cluster
P1 (2,10)	Distance to C1=0 (same point), C2=3.61, C3=8.06	**C1**	Cluster 1
P2 (2,5)	C1=5.0, C2=4.24, C3=3.0	**C3**	Cluster 3
P3 (8,4)	C1=7.62, C2=4.12, C3=8.06	**C2**	Cluster 2
P4 (5,8)	C1=3.61, C2=0.0, C3=7.28	**C2**	Cluster 2
P5 (7,5)	C1=7.21, C2=3.16, C3=7.07	**C2**	Cluster 2
P6 (6,4)	C1=7.81, C2=4.47, C3=5.10	**C2**	Cluster 2
P7 (1,2)	C1=8.06, C2=9.22, C3=0.0	**C3**	Cluster 3
P8 (4,9)	C1=2.24, C2=1.0, C3=7.21	**C2**	Cluster 2
P9 (3,3)	C1=7.07, C2=5.0, C3=2.24	**C3**	Cluster 3

Thus, after Iteration 1:

- **Cluster 1: {P1}**
- **Cluster 2: {P3, P4, P5, P6, P8}**
- **Cluster 3: {P2, P7, P9}**

Recompute centroids (after Iteration 1)

Compute the mean (average) of X and Y for points in each cluster.

- New C1 = mean of {P1} = (2, 10)
- New C2 = mean of {P3(8,4), P4(5,8), P5(7,5), P6(6,4), P8(4,9)}
    - Xmean = (8+5+7+6+4)/5 = 30/5 = 6.0
    - Ymean = (4+8+5+4+9)/5 = 30/5 = 6.0
      → New C2 = (6.0, 6.0)
- New C3 = mean of {P2(2,5), P7(1,2), P9(3,3)}
    - Xmean = (2+1+3)/3 = 6/3 = 2.0
    - Ymean = (5+2+3)/3 = 10/3 ≈ 3.333333...
      → New C3 = (2.0, 3.333333...)

Iteration 2 — Reassign using new centroids

Centroids now: C1= (2,10), C2 = (6,6), C3=(2,3.3333)

Compute distances and assign:

Point	d to C1	d to C2	d to C3	Assigned cluster
P1 (2,10)	0.00	5.39	6.90	C1
P2 (2,5)	5.00	4.12	1.67	C3
P3 (8,4)	7.62	2.83	6.08	C2
P4 (5,8)	3.61	2.24	5.70	C2
P5 (7,5)	7.21	1.41	5.10	C2
P6 (6,4)	7.81	2.24	4.47	C2
P7 (1,2)	8.06	6.71	1.67	C3
P8 (4,9)	2.24	3.16	5.85	C1
P9 (3,3)	7.07	3.61	0.67	C3

Cluster membership after Iteration 2:

- **Cluster 1 (C1): {P1, P8}**
- **Cluster 2 (C2): {P3, P4, P5, P6}**
- **Cluster 3 (C3): {P2, P7, P9}**

Recompute centroids (after Iteration 2)

- New C1 = mean of {P1(2,10), P8(4,9)}
  - Xmean = (2+4)/2 = 3.0
  - Ymean = (10+9)/2 = 9.5
    → C1 = (3.0, 9.5)
- New C2 = mean of {P3(8,4), P4(5,8), P5(7,5), P6(6,4)}
  - Xmean = (8+5+7+6)/4 = 26/4 = 6.5
  - Ymean = (4+8+5+4)/4 = 21/4 = 5.25
    → C2 = (6.5, 5.25)
- New C3 = mean of {P2(2,5), P7(1,2), P9(3,3)} (unchanged)
  → C3 = (2.0, 3.333333...) = (2, 3.33)

Iteration 3 — Reassign using centroids C1=(3.0,9.5), C2=(6.5,5.25), C3=(2.0,3.3333)

Point	**d to C1**	**d to C2**	**d to C3**	**Assigned cluster**
P1 (2,10)	1.12	5.39	6.90	C1
P2 (2,5)	4.72	4.80	1.67	C3
P3 (8,4)	7.28	1.30	6.08	C2
P4 (5,8)	1.80	2.96	5.70	C1
P5 (7,5)	6.10	1.75	5.10	C2
P6 (6,4)	6.40	1.25	4.47	C2
P7 (1,2)	8.79	6.10	1.67	C3
P8 (4,9)	0.50	3.97	5.85	C1
P9 (3,3)	6.53	3.20	0.67	C3

Cluster membership after Iteration 3:

- **Cluster 1 (C1): {P1, P4, P8}**
- **Cluster 2 (C2): {P3, P5, P6}**
- **Cluster 3 (C3): {P2, P7, P9}**

Recompute centroids (after Iteration 3)

- New C1 = mean of {P1(2,10), P4(5,8), P8(4,9)}
  - Xmean = (2+5+4)/3 = 11/3 = 3.6666667
  - Ymean = (10+8+9)/3 = 27/3 = 9.0
    → C1 = (3.6667, 9.0)
- New C2 = mean of {P3(8,4), P5(7,5), P6(6,4)}
  - Xmean = (8+7+6)/3 = 21/3 = 7.0
  - Ymean = (4+5+4)/3 = 13/3 ≈ 4.3333333
    → C2 = (7.0, 4.3333)
- New C3 = mean of {P2(2,5), P7(1,2), P9(3,3)} (unchanged)
  → C3 = (2.0, 3.3333333)

Iteration 4 — Reassign using centroids C1=(3.6667,9.0), C2=(7.0,4.3333), C3=(2.0,3.3333)

Compute distances (summary):

- After computing distances, each point remains assigned as in Iteration 3:

Cluster membership after Iteration 4:

- Cluster 1: {P1, P4, P8}
- Cluster 2: {P3, P5, P6}
- Cluster 3: {P2, P7, P9}

Because assignments did **not change**, the algorithm has **converged**.

Final result (clusters and centroids)

Cluster A (Centroid ≈ (3.6667, 9.0))
Points: P1 (2,10), P4 (5,8), P8 (4,9)

Cluster B (Centroid ≈ (7.0, 4.3333))
Points: P3 (8,4), P5 (7,5), P6 (6,4)

Cluster C (Centroid ≈ (2.0, 3.3333))
Points: P2 (2,5), P7 (1,2), P9 (3,3)

Interpretation (business/analytical)

- **Cluster A** groups the **high-Y values** (customers with high "Y" metric such as high spending) — good candidates for premium offers.
- **Cluster B** groups **medium X, low Y** (e.g., medium income but moderate purchases).
- **Cluster C** groups **low X and low Y** (low-income/low-spend customers).

K-Means successfully partitioned the dataset into three coherent groups that can be used for targeted strategies (marketing, inventory, personalization).

Problem on Regression

calculate a and b by solving the normal equations of the linear regression curve.

x	y	x^2	xy
8	11	64	88
5	10	25	50
4	4	16	16
6	8	36	48
7	9	49	63
9	13	81	117
10	15	100	150
3	6	9	18
2	12	4	24
12	7	144	84

From the above table

- n=10 , ∑ x = 66 , ∑ y = 95 , ∑ xy =1186 , ∑ x^2 = 528
- Now the normal equations become :
- 95 = 10*b + 66a
- 1186 = 66*b + 528a
- By solving the above two euations we get a = 6.05 and b = -30.429
- The linear regression equation is y = -30.429 + 6.05 x.

CLUSTER ANALYSIS (CLUSTERING)

It is an **unsupervised learning** technique used in data mining to group a set of objects into clusters (groups) such that **objects in the same cluster are more similar to each other** than to those in other clusters. It helps uncover hidden patterns, relationships, or structures in data without predefined labels.

The goal is to ensure that data points within a cluster are more similar to each other than to those in other clusters. For example, in e-commerce retailers use clustering to group customers based on their purchasing habits. If one group frequently buys fitness gear while another prefers electronics. This helps companies to give personalized recommendations and improve customer experience. It is useful for:

1. **Scalability:** It can efficiently handle large volumes of data.
2. **High Dimensionality:** Can handle high-dimensional data.
3. **Adaptability to Different Data Types:** It can work with numerical data like age, salary and categorical data like gender, occupation.
4. **Handling Noisy and Missing Data:** Usually, datasets contain missing values or inconsistencies and clustering can manage them easily.
5. **Interpretability:** Output of clustering is easy to understand and apply in real-world scenarios.

Points to Remember:

One group is treated as a cluster of data objects

- In the process of cluster analysis, the first step is to partition the set of data into groups with the help of data similarity, and then groups are assigned to their respective labels.
- The biggest advantage of clustering over-classification is it can adapt to the changes made and helps single out useful features that differentiate different group

Characteristics of Clustering

1. **Unsupervised Learning:** No predefined output variable — clusters are formed based on data similarity.
2. **Similarity-based Grouping:** Objects are grouped using distance or similarity measures (e.g., Euclidean distance, Manhattan distance).
3. **Intra-cluster similarity (high):** Items in the same cluster are very similar.
4. **Inter-cluster similarity (low):** Items in different clusters are as different as possible.
5. **Scalability:** Should efficiently handle large datasets.
6. **Interpretability:** Resulting clusters should be meaningful and easy to interpret.
7. **Automatic Classification:** Data points are automatically grouped without human intervention.

Business Applications of Clustering

Domain	Use Case
Marketing & Customer Segmentation	Identify customer groups based on purchasing behavior.
Retail & E-commerce	Product recommendation and basket analysis.
Banking & Finance	Credit scoring, fraud detection, and customer profiling.
Healthcare	Group patients with similar symptoms or medical history.
Insurance	Risk assessment and claim pattern analysis.
Telecom	Identifying churn segments and usage patterns.

Uses of cluster analysis:

- It is widely used in many applications such as image processing, data analysis, and pattern recognition.
- It helps marketers to find the distinct groups in their customer base and they can characterize their customer groups by using purchasing patterns.
- It can be used in the field of biology, by deriving animal and plant taxonomies and identifying genes with the same capabilities.
- It also helps in information discovery by classifying documents on the web.

Distance Metrics

Distance metrics are simple mathematical formulas to figure out how similar or different two data points are. Type of distance metrics we choose plays a big role in deciding clustering results. Some of the common metrics are:

- **Euclidean Distance:** It is the most widely used distance metric. It is defined as the distance between two points in Euclidean space. To find the distance between two points, the length of the line segment that connects the two points should be measured.
- It is like **measuring the straightest and shortest path between two points**. This **metric is** widely utilized in various fields such as machine learning, data analysis, computer vision, and many more applications of AI and ML.

Euclidean Distance Formula

Consider two points (x1, y1) and (x2, y2) in a 2-dimensional space; the Euclidean Distance between them is given by using the formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Euclidean Distance in 3D

If the two points (x1, y1, z1) and (x2, y2, z2) are in a 3-dimensional space, the Euclidean Distance between them is given by using the formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

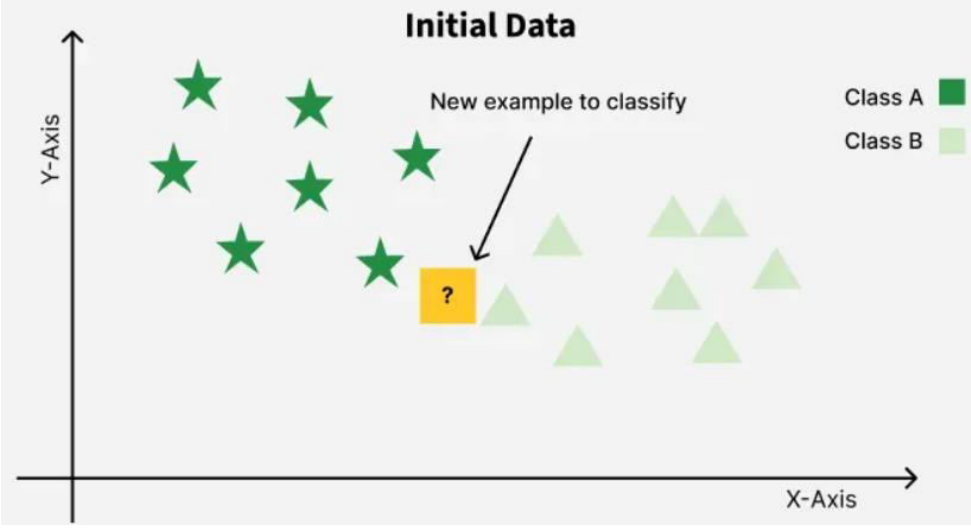
Manhattan Distance: It measures the distance between two points based on grid-like path. It adds the absolute differences between the values.

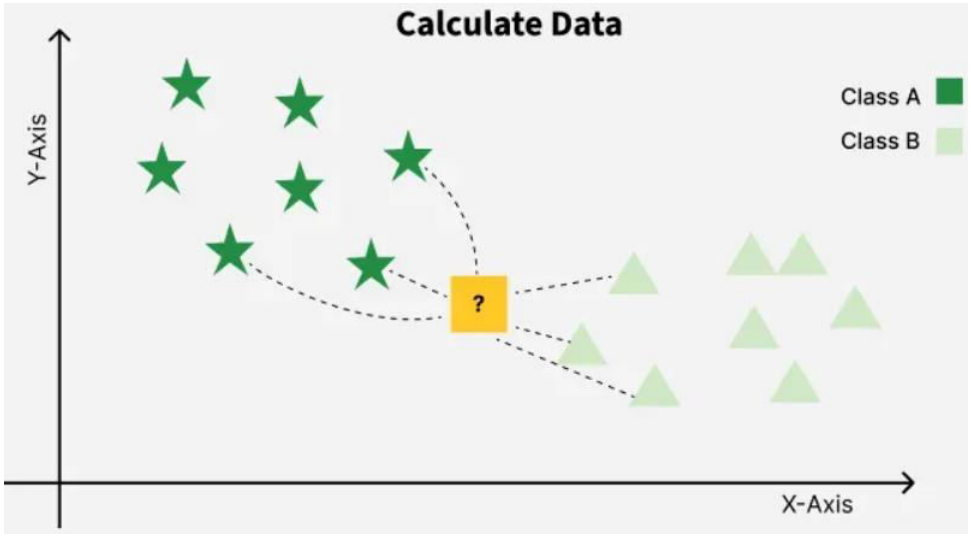
The **Manhattan Distance** between two points **(X1, Y1)** and **(X2, Y2)** is given by **|X1 – X2| + |Y1 – Y2|**. An example is as follows:

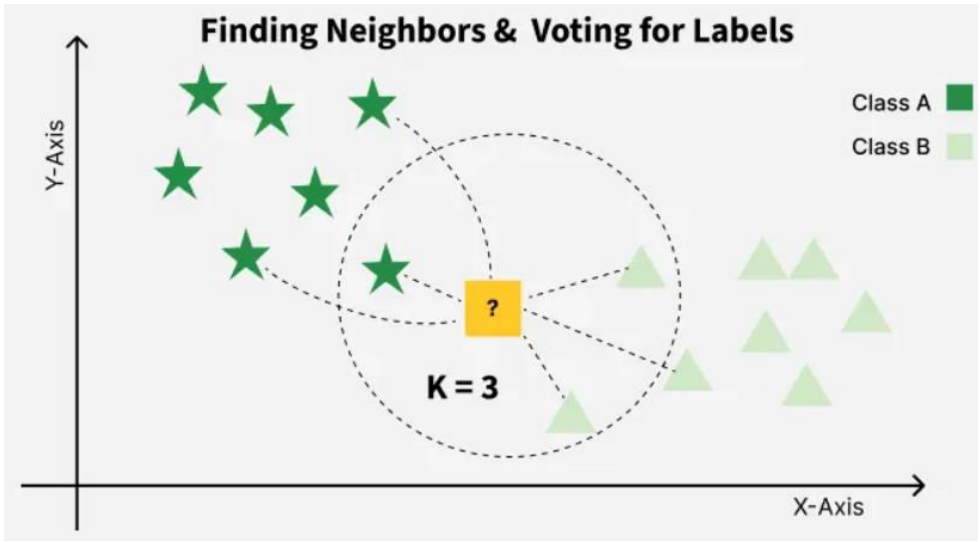
Input: arr[] = {(1, 2), (2, 3), (3, 4)}
Output: 4
Explanation:
The maximum Manhattan distance is found between (1, 2) and (3, 4) i.e.,
|3 - 1| + |4- 2 | = 4.

K-Nearest Neighbour(KNN) Algorithm

KNN is a supervised ML algorithm generally used for classification but can also be used for regression tasks. It works by finding the "k" closest data points (neighbors) to a given input and makes predictions based on the majority class (for classification) or the average value (for regression).



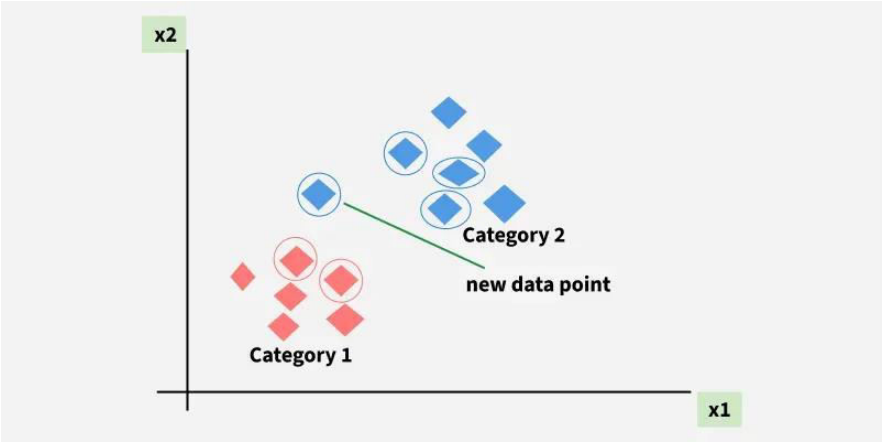




What is the K-Nearest Neighbors Algorithm?

- KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.
- It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data). We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.
- As an example, consider the following table of data points containing two features:

KNN is also called as a lazy learner algorithm because it does not learn from the training set immediately instead it stores the entire dataset and performs computations only at the time of classification. For example, consider the following table of data points containing two features:



The new point is classified as Category 2 because most of its closest neighbours are blue squares. KNN assigns category based on the majority of nearby points.

- The red diamonds represent Category 1 and the blue squares represent Category 2.
- The new data point checks its closest neighbors (circled points).
- Since the majority of its closest neighbors are blue squares (Category 2) KNN predicts the new data point belongs to Category 2.

KNN works by using proximity and majority voting to make predictions.

What is 'K' in K Nearest Neighbour?

In the k-Nearest Neighbours algorithm k is just a number that tells the algorithm how many nearby points or neighbours to look at when it makes a decision.

Example: Imagine you're deciding which fruit it is based on its shape and size. You compare it to fruits you already know.

- If k = 3, the algorithm looks at the 3 closest fruits to the new one.
- If 2 of those 3 fruits are apples and 1 is a banana, the algorithm says the new fruit is an apple because most of its neighbours are apples.

How to choose the value of k for KNN Algorithm?

- The value of k in KNN decides how many neighbours the algorithm looks at when making a prediction. Choosing the right k is important for good results.
- If the data has lots of noise or outliers, using a larger k can make the predictions more stable.
- But if k is too large the model may become too simple and miss important patterns and this is called underfitting.
- So k should be picked carefully based on the data. Rule of thumb: $k \approx \sqrt{n}$

Why KNN Algorithm

- kNN algorithm is a simple, non-parametric, and instance-based machine learning method used for classification and regression. It does not require any assumptions about the underlying data distribution; can also handle both numerical and categorical data.
- Makes predictions based on the similarity of data points in a given dataset. K-NN is less sensitive to outliers compared to other algorithms. It works by finding the K nearest neighbours to a given data point based on a **distance metric**, such as Euclidean distance, which measures the similarity between data points.
- The class or value of the data point is then determined by the majority vote or average of the K neighbours. This approach allows algorithm to adapt to different patterns and make predictions based on the local structure of the data.

Steps in kNN using Euclidean Distance

1. **Compute Euclidean distance** between query point and all points in the dataset.
2. **Sort the distances** in ascending order.
3. **Select the k closest points** (nearest neighbors).
4. **Classify (for classification tasks)** by majority voting among the k neighbors
5. **Predict** by averaging the values of the k nearest neighbors

PROBLEM 1 : We have the following dataset with two features (X_1, X_2) and class labels (0 or 1). Classify a query point (3, 3) using k = 3 and Euclidean distance?

Point (X_1, X_2)	Class
(1, 2)	0
(2, 3)	0
(3, 4)	1
(5, 6)	1

Step 1: Compute Euclidean Distance

Euclidean distance between two points A(X_1, X_2) and B(Y_1, Y_2) is

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Calculate distances from (3,3) to each point in the dataset:

1. Distance to (1,2)

$$d = \sqrt{(3-1)^2 + (3-2)^2} = \sqrt{4+1} = \sqrt{5} \approx 2.24$$

2. Distance to (2,3)

$$d = \sqrt{(3-2)^2 + (3-3)^2} = \sqrt{1+0} = \sqrt{1} = 1.00$$

3. Distance to (3,4)

$$d = \sqrt{(3-3)^2 + (3-4)^2} = \sqrt{0+1} = \sqrt{1} = 1.00$$

4. Distance to (5,6)

$$d = \sqrt{(3-5)^2 + (3-6)^2} = \sqrt{4+9} = \sqrt{13} \approx 3.61$$

Step 2: Sort Distances

Point (X_1, X_2)	Class	Distance from (3,3)
(2,3)	0	1.0
(3,4)	1	1.0
(1,2)	0	2.24
(5,6)	1	3.61

Step 3: Select k Nearest Neighbours

For k=3, the closest 3 points are as follows:

(2,3) → Class 0

(3,4) → Class 1

(1,2) → Class 0

Step 4: Select k Nearest Neighbours

Class 0 : 2 votes

Class 1 : 1 vote

Since **Class 0 has the majority**, we classify (3,3) as **Class 0.**

Answer : The **query point (3, 3) belongs to Class 0.**

EXAMPLE 2 : We'll classify a new 2-D point $x_0 = (4,4)$ using kNN with k = 3 and also with k=5. We will compute distances digit-by-digit to avoid mistakes.

ID	x1	x2	Class
A	1	2	Red
B	2	3	Red
C	3	1	Blue
D	6	5	Blue
E	7	7	Blue
F	8	6	Blue

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Step 1 — compute Euclidean distances to each training point

Distance between (4,4) and A (1,2) = √(9 + 4) = √13 = 3.61

Distance between (4,4) and B (2,3) = √(4 + 1) = √5 = 2.24

Distance between (4,4) and C (3,1) = √(1 + 9) = √10 = 3.16

Distance between (4,4) and D (6,5) = √(4 + 1) = √5 = 2.24

Distance between (4,4) and E (7,7) = √(9 + 9) = √18 = 4.24

Distance between (4,4) and F (8,6) = √(16 + 4) = √20 = 4.48

Step 2 — sort neighbors by distance

Sorted distances (nearest first):

Sorted distances (nearest first):

1. B: 2.24 — Class = Red
2. D: 2.24 — Class = Blue
3. C: 3.16 — Class = Blue
4. A: 3.61 — Class = Red
5. E: 4.24 — Class = Blue
6. F: 4.48 — Class = Blue

(Notice B and D are tied at ≈2.23607; handle ties by including both — fine since k will include both.)

Case 1: k = 3 (majority vote)

Top 3 neighbors: B (Red), D (Blue), C (Blue)

Counts:

- Blue: 2 (D, C)
- Red: 1 (B)

Majority → **Blue**. So $\hat{y} = \text{Blue}.$

Case 2: k = 5 (majority vote)

Top 5 neighbors: B (Red), D (Blue), C (Blue), A (Red), E (Blue)

Counts:

- Blue: 3 (D, C, E)
- Red: 2 (B, A)

Majority → **Blue**. So $\hat{y} = \text{Blue}.$

Problem3 :

We had a dataset of six representing **students' study habits and their exam results**. We need to **classify a new student (S_7)** with this data :

Hours of Study = 3, Classes Attended = 7 using k=3 (i.e., look for 3 nearest neighbors).

Student	Hours of Study (X_1)	Number of Classes Attended (X_2)	Result (Y)
S_1	7	7	Pass
S_2	7	4	Pass
S_3	3	4	Fail
S_4	1	4	Fail
S_5	2	3	Fail
S_6	6	6	Pass

SOLUTION : **Step 1: Compute Euclidean Distances from (3,7)**

Student	Coordinates	Distance from S_7 (3,7)	Result
S_1	(7,7)	√((7−3)² + (7−7)²) = √(16 + 0) = **4.00**	Pass
S_2	(7,4)	√((7−3)² + (4−7)²) = √(16 + 9) = **5.00**	Pass
S_3	(3,4)	√((3−3)² + (4−7)²) = √(0 + 9) = **3.00**	Fail
S_4	(1,4)	√((1−3)² + (4−7)²) = √(4 + 9) = **3.61**	Fail
S_5	(2,3)	√((2−3)² + (3−7)²) = √(1 + 16) = **4.12**	Fail
S_6	(6,6)	√((6−3)² + (6−7)²) = √(9 + 1) = **3.16**	Pass

Step 2: Sort by Distance (Nearest to Farthest)

Rank	Student	Distance	Result
1	S_3	3.00	Fail
2	S_6	3.16	Pass
3	S_4	3.61	Fail
4	S_1	4.00	Pass
5	S_5	4.12	Fail
6	S_2	5.00	Pass

Step 3: Select k = 3 Nearest Neighbours

The **3 nearest neighbours** to S_7 are:

Student	Distance	Result
S_3	3.00	Fail
S_6	3.16	Pass
S_4	3.61	Fail

Step 4: Apply Majority Voting Rule

Among the **3 nearest neighbours**:

- **Fail = 2** (S_3 and S_4)
- **Pass = 1** (S_6)

Therefore, the **predicted result for S_7 = Fail**

Step 5: Interpretation

This means: A student who studies 3 hours and attends 7 classes is likely to **Fail**, based on the patterns learned from historical data.

Advantages of kNN

1. Simple and intuitive (no complex model training).
2. Effective for small datasets.
3. Works well when decision boundaries are irregular.

Limitations

1. Computatssionally expensive for large datasets (distance must be computed for all points).
2. Sensitive to irrelevant or scaled features.
3. Performance depends on the right choice of **k** and **distance metric**.

Pros

- Simple to understand and implement.
- No parametric assumptions about data distribution.
- Naturally handles multi-class problems.
- Flexible: choice of distance and weighting.

Cons

- Slow at prediction time for large datasets (unless optimized).
- Memory intensive (store entire training set).
- Sensitive to irrelevant features and feature scaling.
- Performance deteriorates in high dimensions.

Uses / Applications of KNN Rule in business and industry

- **Customer classification**: label a new customer into segments based on past customers.
- **Credit scoring**: nearest historical borrowers to estimate default risk.
- **Recommendation (cold-start)**: content/item similarity (kNN on item features).
- **Fraud detection**: compare transaction to neighbors to spot anomalies.
- **Medical diagnosis**: classify patient condition from clinical measurements.
- **Image recognition (classical)**: nearest-neighbor on feature descriptors.

Example: Table shows the number of motor registrations in a certain territory for a term of 5 years and the sale of tyres by a firm in that territory

Year	Motor Registrations	No. of Tyres Sold
1	600	1,250
2	630	1,100
3	720	1,300
4	750	1,350
5	800	1,500

Find the regression equation to estimate the sale of tyres when motor registration is known.

Estimate sale of tyres when registration is 850.

Solution: We take registrations as *X* and tyre sales as *Y.* To find line of *Y* on *X*.

X	Y	$d_x = X - \bar{X}$	$d_y = Y - \bar{Y}$	d_x^2	$d_x d_y$
600	*1,250*	*-100*	*-50*	*10,000*	*5,000*
630	*1,100*	*-70*	*-200*	*4,900*	*14,000*
720	*1,300*	*20*	*0*	*400*	*0*
750	*1,350*	*50*	*50*	*2,500*	*2,500*
800	*1,500*	*100*	*200*	*10,000*	*20,000*
$\sum X = 3,500$	$\sum Y = 6,500$	$\sum d_x = 0$	$\sum d_y = 0$	$\sum d_x^2 = 27,800$	$\sum d_x d_y = 41,500$

$$\bar{X} = \frac{\sum X}{N} = \frac{3,500}{5} = 700 \quad \text{and} \quad \bar{Y} = \frac{\sum Y}{N} = \frac{6,500}{5} = 1,300$$

b_{yx} = Regression coefficient of Y on X

$$b_{yx} = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sum (X - \bar{X})^2} = \frac{\sum d_x d_y}{\sum d_x^2} = \frac{4,1500}{2,7800} = 1.4928$$

The regression line of Y on X is given by the equation:

$$Y - \bar{Y} = b_{yx} (X - \bar{X})$$

or $$Y - 1300 = 1.4928 (X - 700)$$

$$Y = 1.4928 X + 255.04$$

When $X = 850$, the value of Y can be calculated from the above equation, by putting $X = 850$

***Y* = (1.4928 *x* 850) + 255. 04 = 1523.92 = 1524 tyres**

Example: For some bivariate data, the following results were obtained

Mean value of variable X = 53.2

Mean value of variable Y = 27.9

Regression coefficient of Y on X = - 1.5

Regression coefficient of X on Y = - 0.2

What is the most likely value of Y, when $X = 60$?

What is the coefficient of correlation between X and Y?

$\overline{X}$ = 53.2 $\overline{Y}$ = 27.9

b_{yx} = -1.5 b_{xy} = -0.2

To obtain value of Y for $X = 60$, we establish the regression line of Y on X,

$$Y - \overline{Y} = b_{yx}\ (X - \overline{X})$$

$$Y - 27.9 = -1.5\ (X - 53.2)$$

or $$Y = -1.5X + 107.7$$

Putting value of $X = 60$, we obtain

$$Y = -1.5 \times 60 + 107.7$$

$$= 17.7$$

Coefficient of correlation between *X, Y* is given by $r^2 = b_{yx} \cdot b_{xy}$

$$r = \pm\sqrt{0.3} = \pm 0.5477$$

Since both regression coefficients are negative, we assign negative value to correlation coefficient r, and so r = -0.5477

Newton's Method in Optimization

Newton's Method (or the Newton–Raphson method) is a **second-order iterative optimization technique** used to find **stationary points** (minima, maxima, or saddle points) of a real-valued differentiable function $f(x)$.

It extends the 1D Newton–Raphson root-finding method to optimization problems by finding where the **gradient** (first derivative) becomes zero.

Objective:

We want to find x^* such that:

$$\nabla f(x^*) = 0$$

where

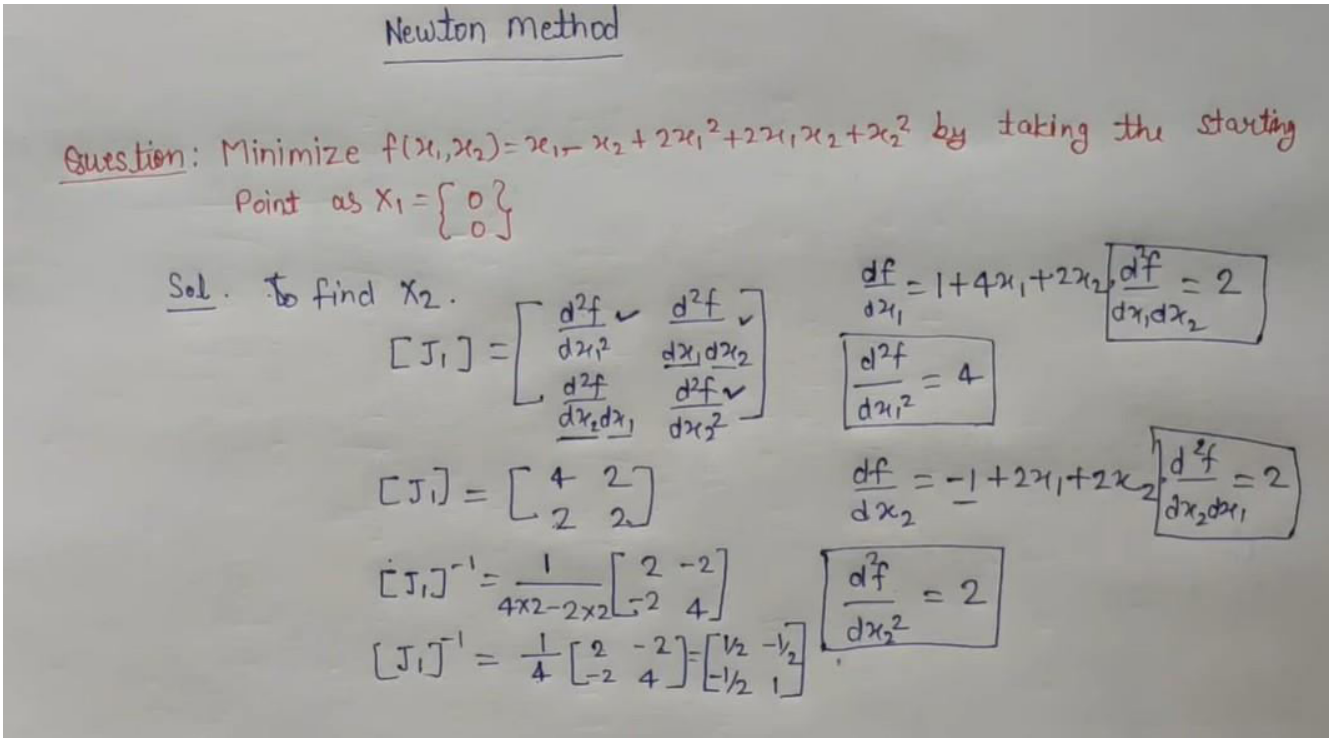
- $\nabla f(x)$ = gradient vector of $f(x)$,
- $\nabla^2 f(x)$ = Hessian matrix (matrix of second derivatives).

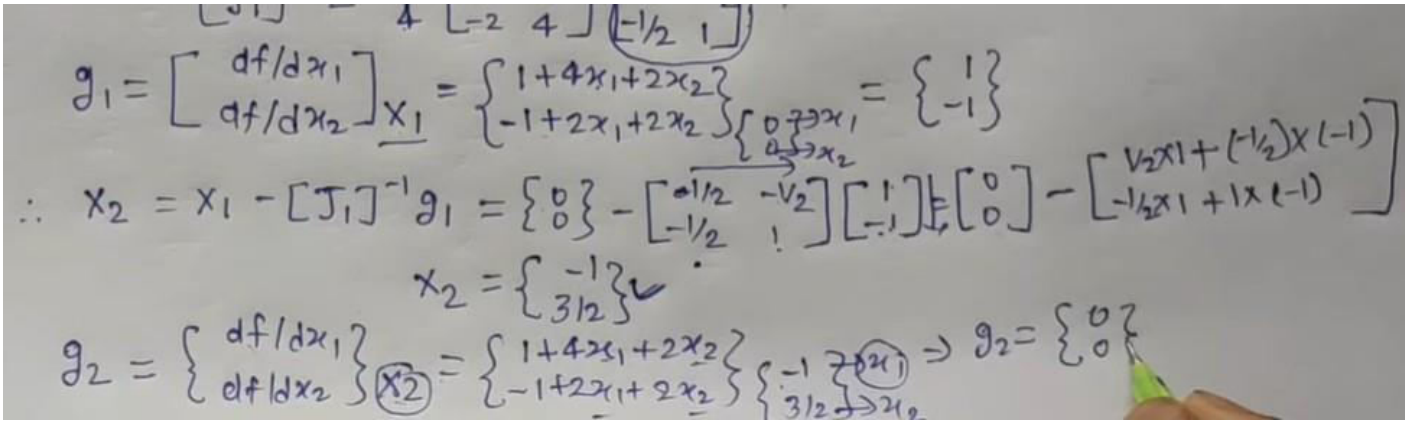
Algorithm

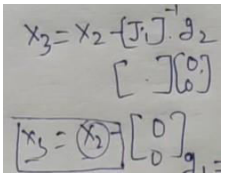
1. **Initialize:** Choose a starting point x_0.
2. **Compute gradient:** $g_k = \nabla f(x_k)$.
3. **Compute Hessian:** $H_k = \nabla^2 f(x_k)$.
4. **Compute search direction:** $d_k = -H_k^{-1} g_k$.
5. **Update:** $x_{k+1} = x_k + d_k$.
6. **Check convergence:** If $||g_{k+1}|| < \epsilon$, stop; else repeat.

Disadvantages of Newton's Method

Disadvantage	Explanation
1. Requires Hessian computation	The Hessian ((n \times n) matrix) must be computed and inverted — expensive for large (n).
2. May not converge	If the Hessian is not positive definite (saddle point or maximum), the step can move away from minimum.
3. Sensitive to initial guess	Poor starting point can lead to divergence or convergence to the wrong stationary point.
4. High computational cost	Computing and inverting the Hessian costs (O(n^3)).
5. Not suitable for non-smooth functions	Requires continuous second derivatives.
6. Step may overshoot	If the step size is too large, the quadratic approximation fails — often a line search or damping factor is added.





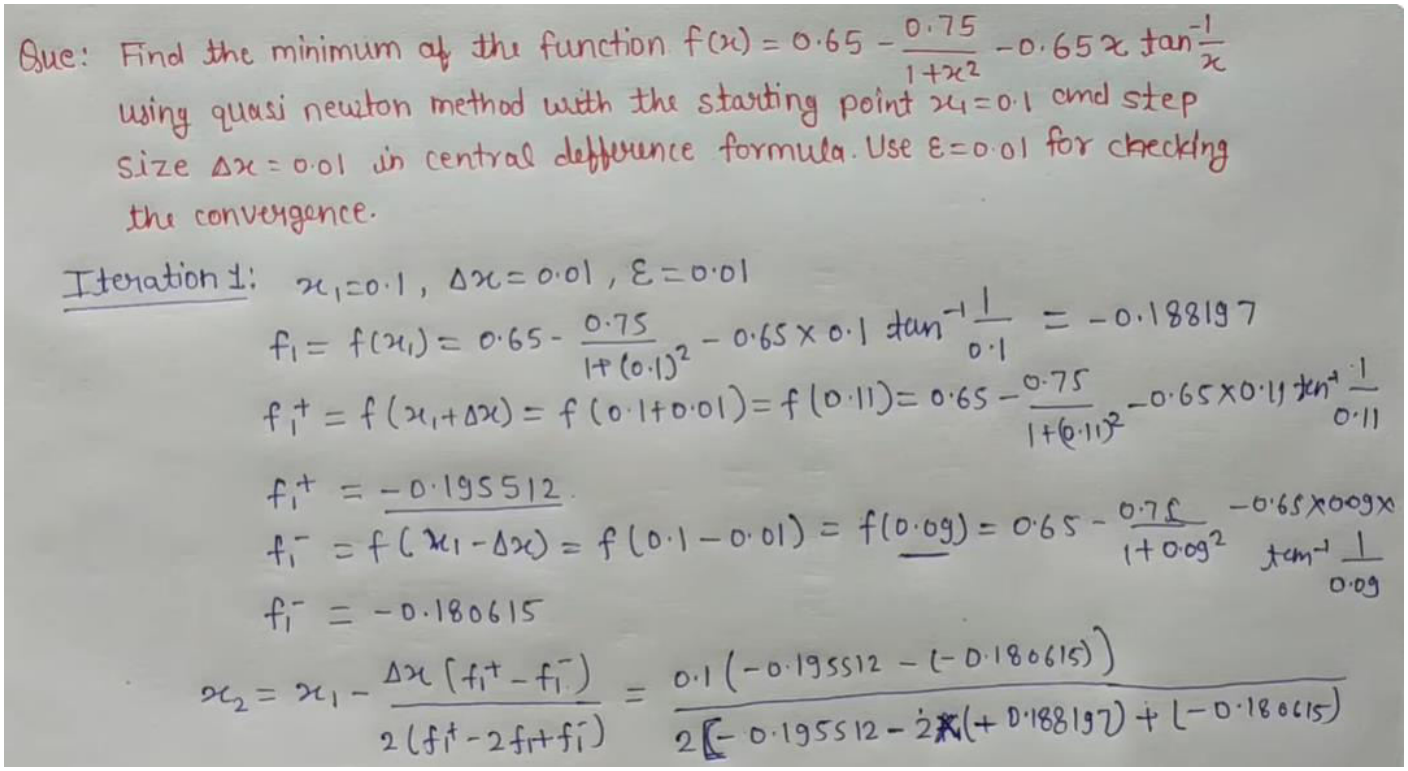


Newton vs. Gradient Descent

Feature	Gradient Descent	Newton's Method
Uses	Gradient only	Gradient + Hessian
Step Direction	Negative gradient	Inverse Hessian × gradient
Convergence Rate	Linear	Quadratic (faster near optimum)
Cost per Iteration	Low	High (Hessian inversion)
Suitable for	Large-scale, simple problems	Smaller, well-behaved quadratic problems

Quasi Newton Method

- Quasi-Newton methods are optimization algorithms that find minima or maxima of functions by approximating the Hessian matrix of second derivatives.
- Unlike Newton's method, they avoid the computational cost of calculating the true Hessian, instead updating an approximation using gradient and position information from previous steps.
- This makes them more efficient for large-scale problems while still achieving superlinear convergence.



Que: Find the minimum of the function $f(x) = 0.65 - \frac{0.75}{1+x^2} - 0.65x \tan^{-1}\frac{1}{x}$ using quasi newton method with the starting point $x_1 = 0.1$ and step size $\Delta x = 0.01$ in central defference formula. Use $\varepsilon = 0.01$ for checking the convergence.

Iteration 1: $x_1 = 0.1$, $\Delta x = 0.01$, $\varepsilon = 0.01$

$f_1 = f(x_1) = 0.65 - \frac{0.75}{1+(0.1)^2} - 0.65 \times 0.1 \tan^{-1}\frac{1}{0.1} = -0.188197$

$f_1^+ = f(x_1+\Delta x) = f(0.1+0.01) = f(0.11) = 0.65 - \frac{0.75}{1+(0.11)^2} - 0.65\times 0.11 \tan^{-1}\frac{1}{0.11}$

$f_1^+ = -0.195512$

$f_1^- = f(x_1-\Delta x) = f(0.1-0.01) = f(0.09) = 0.65 - \frac{0.75}{1+0.09^2} - 0.65\times 0.09 \times \tan^{-1}\frac{1}{0.09}$

$f_1^- = -0.180615$

$x_2 = x_1 - \frac{\Delta x (f_1^+ - f_1^-)}{2(f_1^+ - 2f_1 + f_1^-)} = \frac{0.1(-0.195512 - (-0.180615))}{2[-0.195512 - 2(-0.188197) + (-0.180615)]}$

So, we have $x_2 = 0.377882$

$$|f'(x_2)| = \left|\frac{f_2^+ - f_2^-}{2\Delta x}\right| = 0.137300 > \varepsilon = 0.01$$

$f_2^+ = f(x_2 + \Delta x)$
$= f(0.377882 + 0.01)$
$= f(0.387882)$
$= -0.304662$

$f_2^- = f(x_2 - \Delta x)$
$= f(0.377882 - 0.01)$
$= f(0.367882)$
$= -0.301916$

Iteration 2:

$f_2 = f(x_2) = -0.303368.$

$f_2^+ = 0.304662$

$f_2^- = -0.301916.$

$$x_3 = x_2 - \frac{\Delta x(f_2^+ - f_2^-)}{2(f_2^+ - 2f_2 + 2f_2^-)} = 0.465390$$

Convergence check.

$$|f'(x_3)| = \left|\frac{f_3^+ - f_3^-}{2\Delta x}\right| = 0.0177 > \varepsilon = 0.01$$

$f_3^+ = f(x_3 + \Delta x)$
$= f(0.465390 + 0.01)$
$= f(0.475390)$
$= -0.310004$

$f_3^- = f(x_3 - \Delta x)$
$= f(0.465390)$
$= -0.455390$

Iteration 3:

$$f_3 = f(x_3) = 0.65 - \frac{0.75}{1 + 0.465392} - 0.65 \times 0.46539 \times \tan^{-1}\frac{1}{0.46539}$$

$f_3 = -0.309885$

$f_3^+ = -0.310004, \quad f_3^- = -0.309650$

$$x_4 = x_3 - \frac{\Delta x(f_3^+ - f_3^-)}{2(f_3^+ - 2f_3 + f_3^-)} = 0.480600$$

Convergence check.

$$|f'(x_4)| = \left|\frac{f_4^+ - f_4^-}{2\Delta x}\right| = 0.000350 < \varepsilon = 0.01$$

$f_4^+ = f(x_4 + \Delta x)$
$= f(0.490600)$
$= -0.3099688$

$f_4^- = f(x_4 - \Delta x)$
$= f(0.47060)$
$= -0.3099615$

· process has converged we take the optimum solution as $x^* \approx x_4 = 0.480600.$

Concept of Regression Tree

A Regression Tree is a type of Decision Tree used when the target (output) variable is continuous (not categorical).

It divides the data into smaller and smaller regions so that the value of the dependent variable (Y) within each region is as homogeneous as possible.

How It Works

1. Start with all the training data.
2. At each node, select the variable and the split point that minimizes the sum of squared errors (SSE) or variance within the resulting groups.
3. Repeat splitting recursively until a stopping condition is met (e.g., minimum number of samples, or no significant improvement).
4. The predicted value for each terminal node (leaf) is the mean of the target variable in that region.

Mathematical Criterion for Split

For a split based on variable X_j at split point s:

$$R_1(j,s) = \{X | X_j \leq s\}, \quad R_2(j,s) = \{X | X_j > s\}$$

The cost function is:

$$C(j,s) = \sum_{x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2$$

We choose (j^*, s^*) that **minimizes** $C(j,s)$.

Dataset

We want to predict the **Sales (Y)** based on the **Advertising Spend (X)** (in ₹ lakh):

Observation	X (Advertising)	Y (Sales)
1	2	4
2	4	6
3	6	8
4	8	10
5	10	11

We will build a **simple regression tree** using one variable X.

Step 1: Find Possible Splits

The possible split points are midpoints between X values:

$$s = 3, 5, 7, 9$$

Step 2: For each split, compute SSE

We'll calculate for each split:

$$SSE = \sum (y_i - \bar{y}_{left})^2 + \sum (y_i - \bar{y}_{right})^2$$

Split 1: s = 3

Left (X ≤ 3): Y = [4] → mean = 4

Right (X > 3): Y = [6, 8, 10, 11] → mean = 8.75

SSE =

Left: $(4-4)^2 = 0$

Right: $(6-8.75)^2 + (8-8.75)^2 + (10-8.75)^2 + (11-8.75)^2$

$= 7.56 + 0.56 + 1.56 + 5.06 =$ **14.74**

→ **Total SSE = 14.74**

Split 2: s = 5

Left ($X \le 5$): Y = [4, 6] → mean = 5

Right ($X > 5$): Y = [8, 10, 11] → mean = 9.67

SSE =

Left: $(4-5)^2 + (6-5)^2 = 1 + 1 = 2$

Right: $(8-9.67)^2 + (10-9.67)^2 + (11-9.67)^2 = 2.78 + 0.11 + 1.78 = 4.67$

→ **Total SSE = 2 + 4.67 = 6.67**

Split 3: s = 7

Left ($X \le 7$): Y = [4, 6, 8] → mean = 6

Right ($X > 7$): Y = [10, 11] → mean = 10.5

SSE =

Left: $(4-6)^2 + (6-6)^2 + (8-6)^2 = 4 + 0 + 4 = 8$

Right: $(10-10.5)^2 + (11-10.5)^2 = 0.25 + 0.25 = 0.5$

→ **Total SSE = 8 + 0.5 = 8.5**

Split 4: s = 9

Left ($X \le 9$): Y = [4, 6, 8, 10] → mean = 7

Right ($X > 9$): Y = [11] → mean = 11

SSE =

Left: $(4-7)^2 + (6-7)^2 + (8-7)^2 + (10-7)^2 = 9 + 1 + 1 + 9 = 20$

Right: $(11-11)^2 = 0$

→ **Total SSE = 20**

Step 3: Choose the Best Split

Split	SSE	Decision
s = 3	14.74	—
s = 5	**6.67 (minimum)**	✅ Best
s = 7	8.5	—
s = 9	20	—

Hence, the **best split** is at **X = 5**.

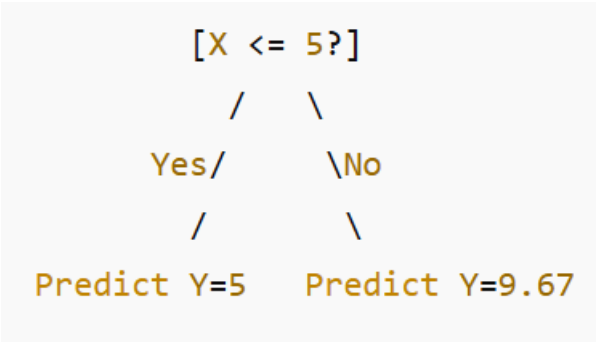
Step 4: Interpret the Tree

- **If X ≤ 5**: predict Y = 5
- **If X > 5**: predict Y = 9.67

Step 5: Predicted values

X	Y (Actual)	Predicted Y
2	4	5
4	6	5
6	8	9.67
8	10	9.67
10	11	9.67

Final Regression Tree



Mean Squared Error (MSE) in Regression Analysis

In regression analysis, the **Mean Squared Error (MSE)** is a commonly used metric to measure how well a regression model fits the data. It represents the **average of the squares of the errors** — that is, the average squared difference between the **actual (observed)** and the **predicted** values. It is used as a **loss function** in regression algorithms (e.g., linear regression, neural networks). During model training, the goal is often to **minimize the MSE**.

If the actual values are $y_1, y_2, \ldots, y_n$ and the corresponding predicted values from the regression model are $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$, then:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where:

- y_i = actual value
- $\hat{y}_i$ = predicted value
- n = number of data points

Interpretation

- **MSE = 0** means a perfect fit (predictions are exactly equal to actual values).
- A **smaller MSE** value indicates a better fit of the regression model to the data.
- A **larger MSE** value indicates poor predictive accuracy — the model's predictions deviate more from actual data.

Observation (i)	Actual Value y_i	Predicted Value $\hat{y}_i$	Error $y_i - \hat{y}_i$	Squared Error $(y_i - \hat{y}_i)^2$
1	10	12	-2	4
2	8	9	-1	1
3	12	11	1	1
4	14	13	1	1

$$\text{MSE} = \frac{4+1+1+1}{4} = \frac{7}{4} = 1.75$$

Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2}$	Square root of MSE; same units as the response variable.
R^2 (Coefficient of Determination)	$1 - \frac{\text{SSR}}{\text{SST}}$	Measures proportion of variance explained by the model.

Advantages of MSE

- Easy to compute and widely used.
- Useful in optimization algorithms (like gradient descent in machine learning).
- Penalizes large errors heavily due to squaring, which helps discourage outliers.

Limitations of MSE

- Sensitive to outliers because errors are squared.
- The value is not in the same units as the original data (since errors are squared).
- May not be intuitive for interpretation compared to MAE.

Usage in Model Evaluation

In regression analysis and machine learning:

- MSE helps compare different models — the one with the lowest MSE is usually preferred.
- It is commonly used as the loss function in training models like **Linear Regression**, **Ridge Regression**, and **Neural Networks**.

Numerical on multiple regression analysis

Problem

A company wants to predict **Sales (Y)** based on two independent variables:

- X_1 = Advertising expenditure (in ₹ lakhs)
- X_2 = Number of salespeople

The following data is collected:

Observation	X_1	X_2	Y (Sales)
1	1	2	2
2	2	1	3
3	3	4	6
4	4	3	7
5	5	5	11

We need to find the regression equation:

$$Y = b_0 + b_1 X_1 + b_2 X_2$$

STEP 1: Compute the required means

$$\bar{X}_1 = \frac{1+2+3+4+5}{5} = 3$$

$$\bar{X}_2 = \frac{2+1+4+3+5}{5} = 3$$

$$\bar{Y} = \frac{2+3+6+7+11}{5} = 5.8$$

STEP 2: Compute the deviations

Obs	X_1	X_2	Y	$(X_1-\bar{X}_1)$	$(X_2-\bar{X}_2)$	$(Y-\bar{Y})$
1	1	2	2	-2	-1	-3.8
2	2	1	3	-1	-2	-2.8
3	3	4	6	0	1	0.2
4	4	3	7	1	0	1.2
5	5	5	11	2	2	5.2

STEP 3: Compute the necessary sums

Quantity	Formula	Calculation	Value
S_{x1x1}	$\Sigma(X_1-\bar{X}_1)^2$	$(-2)^2 + (-1)^2 + 0^2 + 1^2 + 2^2$	10
S_{x2x2}	$\Sigma(X_2-\bar{X}_2)^2$	$(-1)^2 + (-2)^2 + 1^2 + 0^2 + 2^2$	10
S_{x1x2}	$\Sigma(X_1-\bar{X}_1)(X_2-\bar{X}_2)$	$(-2)(-1) + (-1)(-2) + 0(1) + 1(0) + 2(2)$	8
S_{x1y}	$\Sigma(X_1-\bar{X}_1)(Y-\bar{Y})$	$(-2)(-3.8)+(-1)(-2.8)+0(0.2)+1(1.2)+2(5.2)$	21.0
S_{x2y}	$\Sigma(X_2-\bar{X}_2)(Y-\bar{Y})$	$(-1)(-3.8)+(-2)(-2.8)+1(0.2)+0(1.2)+2(5.2)$	17.0

STEP 4: Formulas for b_1 and b_2

For two independent variables:

$$b_1 = \frac{S_{x2x2}S_{x1y} - S_{x1x2}S_{x2y}}{S_{x1x1}S_{x2x2} - S_{x1x2}^2}$$

$$b_2 = \frac{S_{x1x1}S_{x2y} - S_{x1x2}S_{x1y}}{S_{x1x1}S_{x2x2} - S_{x1x2}^2}$$

Compute the denominator:

$$D = S_{x1x1}S_{x2x2} - S_{x1x2}^2 = (10)(10) - (8)^2 = 100 - 64 = 36$$

Now compute numerators

$$\text{Numerator for } b_1 = (10)(21.0) - (8)(17.0) = 210 - 136 = 74$$

$$\text{Numerator for } b_2 = (10)(17.0) - (8)(21.0) = 170 - 168 = 2$$

Hence $b_1 = 74/36 = 2.056$, $b_2 = 2/36 = 0.056$

STEP 5: Compute the intercept b_0

$$b_0 = \bar{Y} - b_1\bar{X}_1 - b_2\bar{X}_2$$

Value of b_0 is = 5.8 – (2.056 *3) – (0.056 * 3) = = 5.8 – 6.336 = - 0.536

Final Regression Equation is Y = - 0.536 + 2.056 X_1 + 0.056 X_2

Interpretation

- When both X_1 and X_2 = 0, predicted sales = **– 0.536** (the intercept).
- For every unit increase in **advertising expenditure (X_1)**, sales increase by 2.056 **units** (holding X_2 constant).
- For every additional **salesperson (X_2)**, sales increase by **0.667 units** (holding X_1 constant).