Module2

Dr. Thyagaraju G S

Contents

1. Informed Search Strategies:
   a. Greedy best-first search,
   b.A*search,
   c. Heuristic functions.
2. Introduction to Machine Learning,
3.Understanding Data

2. 1 Informed Search Strategies

- **Informed Search:** Informed search is a search strategy that utilizes problem-specific knowledge, to find solutions more efficiently. Informed search methods make use of heuristics and evaluation functions to guide the search towards more promising paths.

- **Heuristic Function(h(n))**: It is a heuristic function that provides an estimate of the cost from the current node to the goal node. This heuristic is admissible if it never overestimates the true cost to reach the goal. In other words, **h(n)** is always less than or equal to the actual cost.

- **Actual cost function(g(n)): It is** Cost of the path from the **start node to node n**.It represents the actual cost incurred to reach the current node from the initial node. For the initial node (start node), **g(n)** is usually 0.

- **Evaluation Function (f(n))**: The evaluation function, denoted as **f(n),** is the total estimated cost of the cheapest path from the start node to the goal node that passes through node n. It is the sum of g(n) and h(n): **f(n) = g(n) + h(n).**

2.1.a Greedy Best First Search Algorithm

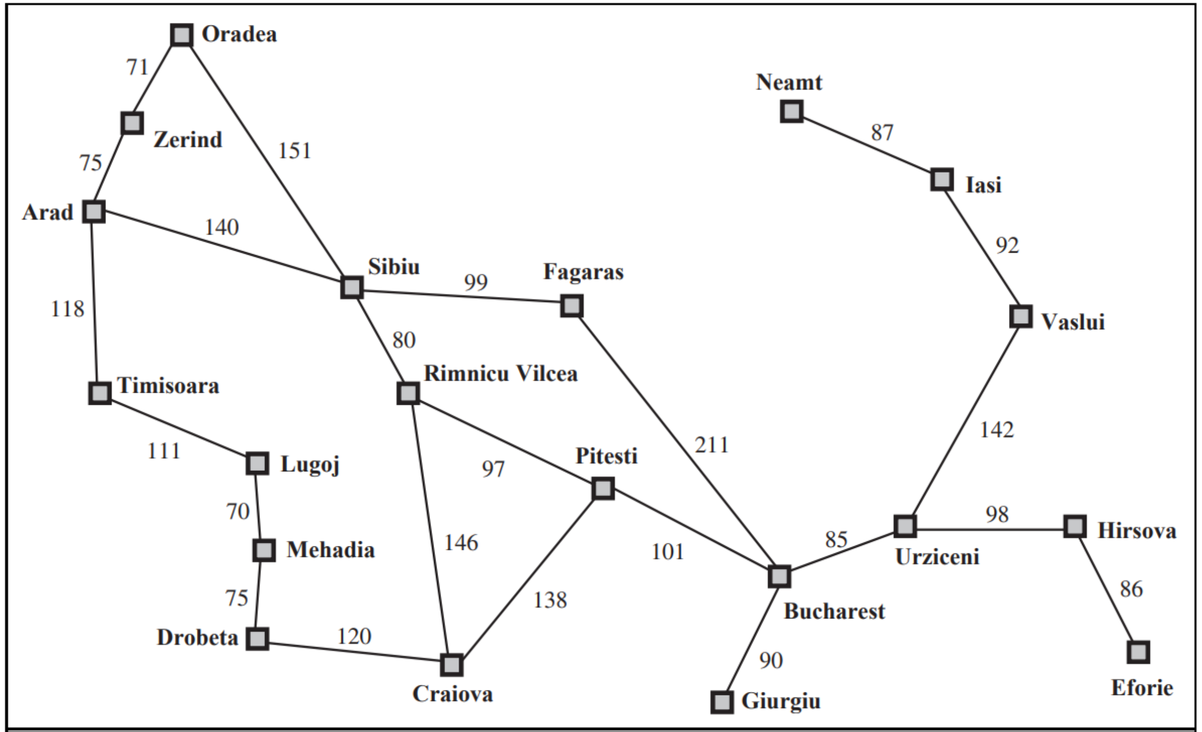
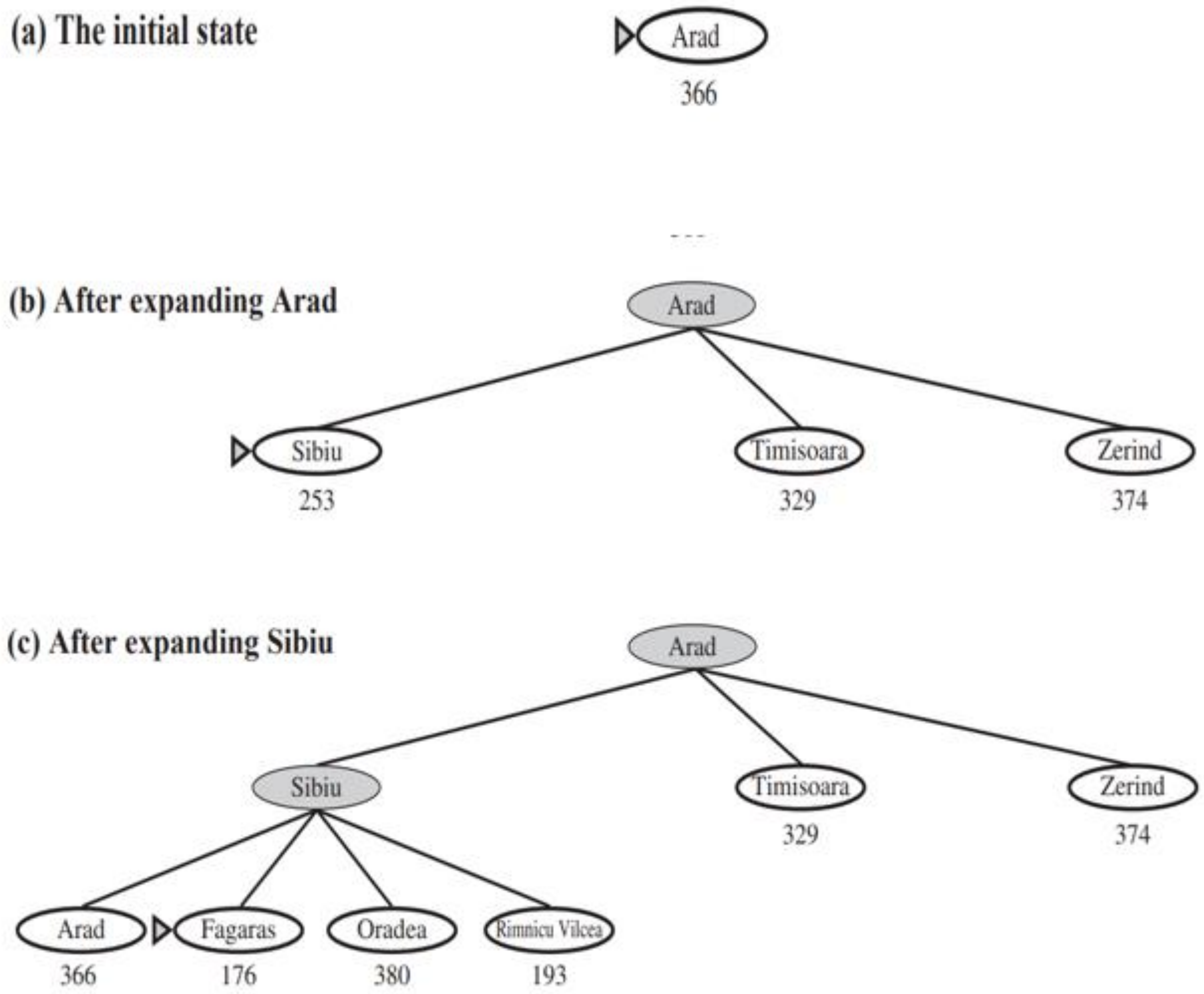


Figure 3.2 A simplified road map of part of Romania.

Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

Figure 3.22 Values of h_{SLD}—straight-line distances to Bucharest.

Figure: Stages in a greedy best-first tree search for Bucharest with the straight-line distance heuristic hSLD . Nodes are labeled with their h-values.



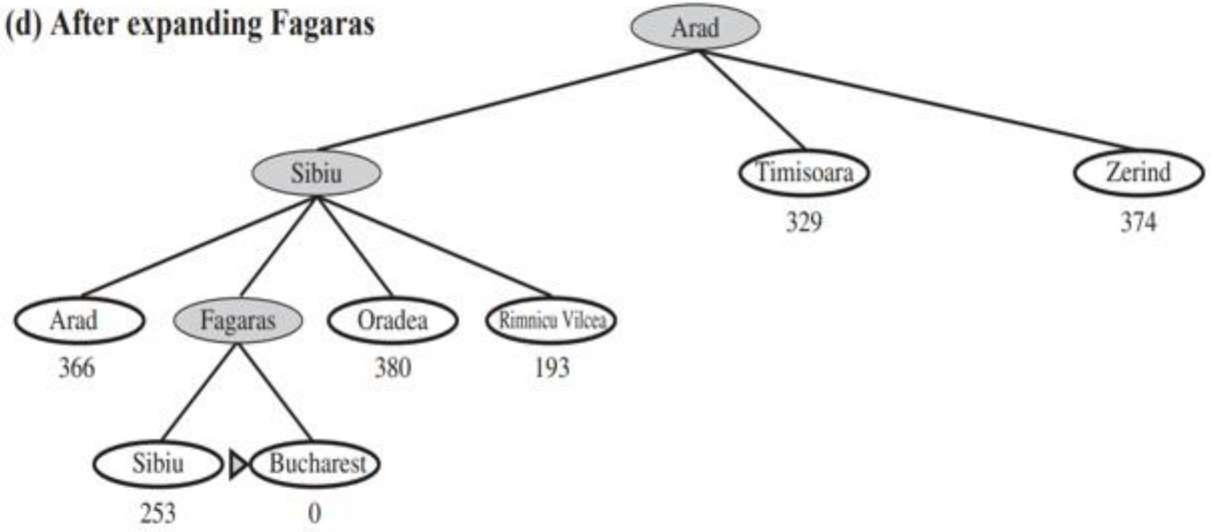


Figure: Stages in a greedy best-first tree search for Bucharest with the straight-line distance heuristic hSLD . Nodes are labeled with their h-values.

Best first search algorithm:

Step 1: Place the starting node into the OPEN list.

Step 2: If the OPEN list is empty, Stop and return failure.

Step 3: Remove the node n, from the OPEN list which has the lowest value of h(n), and places it in the CLOSED list.

Step 4: Expand the node n, and generate the successors of node n.

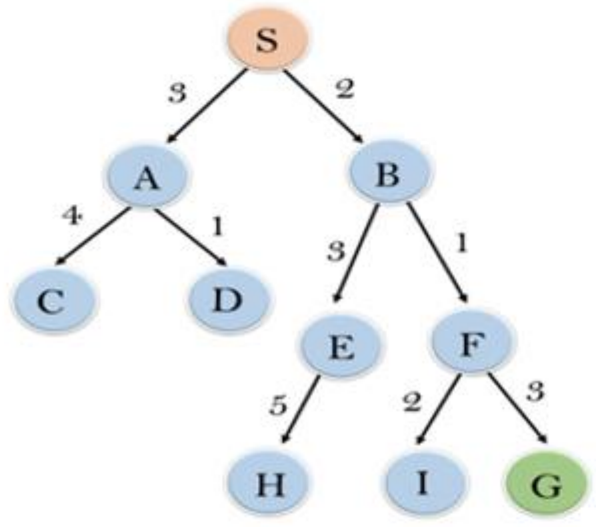
Step 5: Check each successor of node n, and find whether any node is a goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to Step 6.

Step 6: For each successor node, algorithm checks for evaluation function f(n), and then check if the node has been in either OPEN or CLOSED list. If the node has not been in both list, then add it to the OPEN list.

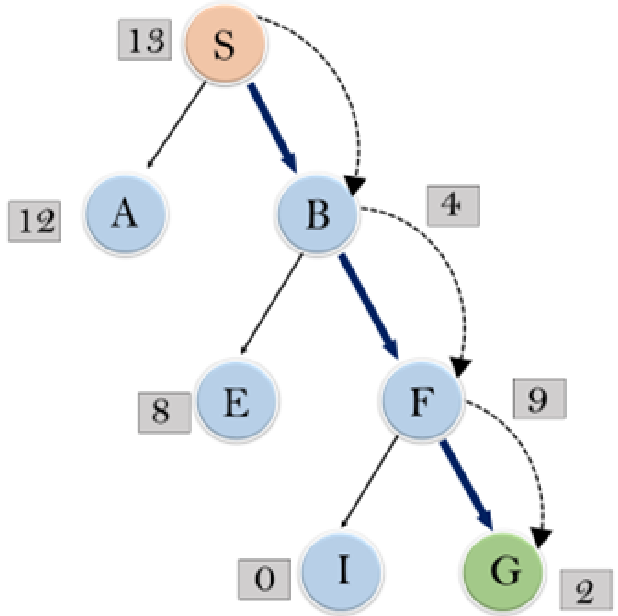
Step 7: Return to Step 2.

Example

Step	OPEN List	CLOSED List	Details
Initialization	[A, B]	[S]	
Iteration1: Expand B	[A]	[S,B]	h(B)<h(A)
Iteration2: Expand F	[E,F,A] [E,A]	[S,B] [S,B,F]	H(F)<H(E),H(A)
Iteration3: Visit Goal G	[I,G,E,A] [I,E,A]	[S,B,F] [S,B,F,G]	H(G)< H(E),H(A),H(I)



node	H (n)
A	12
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

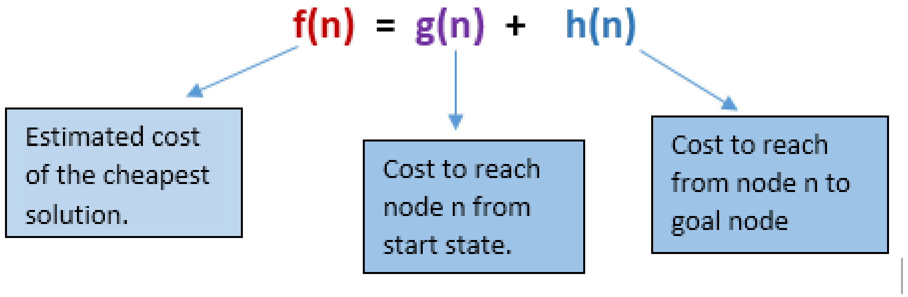


Hence the final solution path will be: **S----> B----->F----> G**

A* Search Algorithm

- A* search is the most commonly known form of best-first search. It uses heuristic function h(n), and cost to reach the node n from the start state g(n). It has combined features of UCS and greedy best-first search, by which it solve the problem efficiently. A* search algorithm finds the shortest path through the search space using the heuristic function. This search algorithm expands less search tree and provides optimal result faster. A* algorithm is similar to UCS except that it uses g(n)+h(n) instead of g(n).

- In A* search algorithm, we use search heuristic as well as the cost to reach the node. Hence we can combine both costs as following, and this sum is called as a **fitness number**.



Algorithm of A* search:

Step1: Place the starting node in the OPEN list.

Step 2: Check if the OPEN list is empty or not, if the list is empty then return failure and stops.

Step 3: Select the node from the OPEN list which has the smallest value of evaluation function (g+h), if node n is goal node then return success and stop, otherwise

Step 4: Expand node n and generate all of its successors, and put n into the closed list. For each successor n', check whether n' is already in the OPEN or CLOSED list, if not then compute evaluation function for n' and place into Open list.

Step 5: Else if node n' is already in OPEN and CLOSED, then it should be attached to the back pointer which reflects the lowest g(n') value.

Step 6: Return to Step 2.

Example1

Let's explore the application of this method to route-finding challenges in Romania for the map given in figure 3.2 .

We will employ the straight-line distance heuristic, denoted as hSLD. Specifically, for our destination in Bucharest, we require knowledge of the straight-line distances to Bucharest, as illustrated in Figure 3.22.

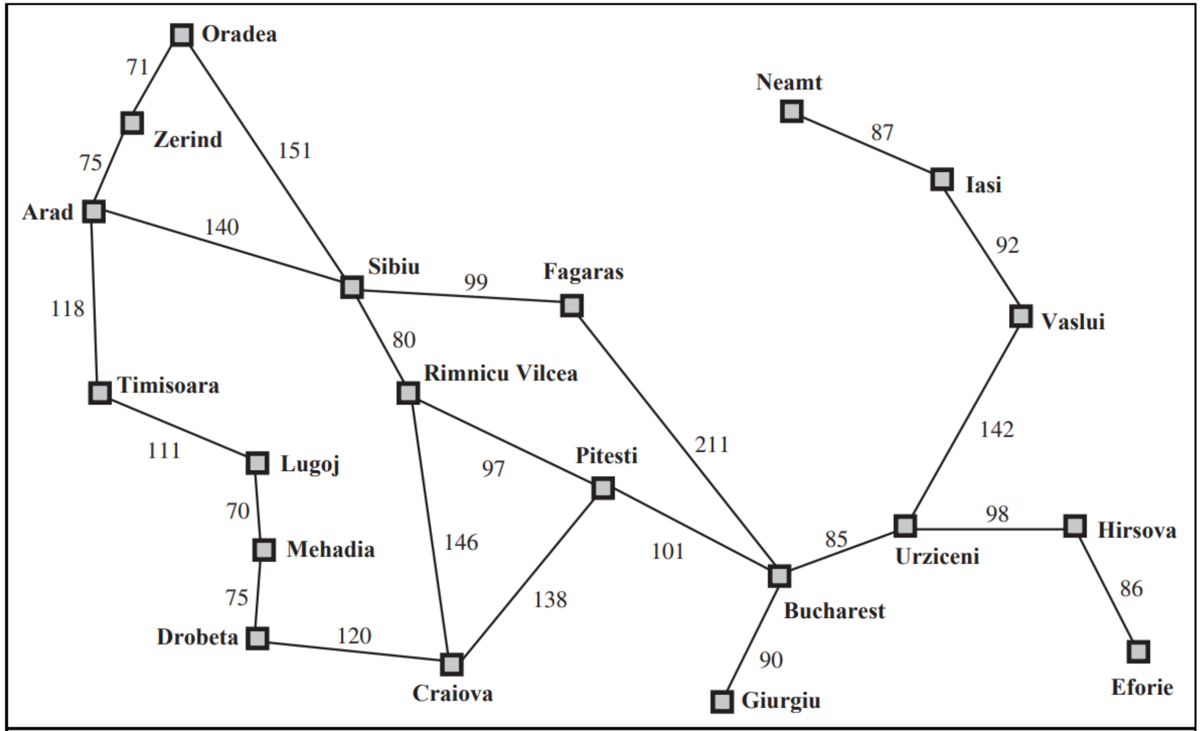


Figure 3.2 A simplified road map of part of Romania.

Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

Figure 3.22 Values of h_{SLD}—straight-line distances to Bucharest.

Figure below illustrates the Stages in **an A∗ search for Bucharest**. Nodes are labeled with f = g + h. The h values are the straight-line distances to Bucharest

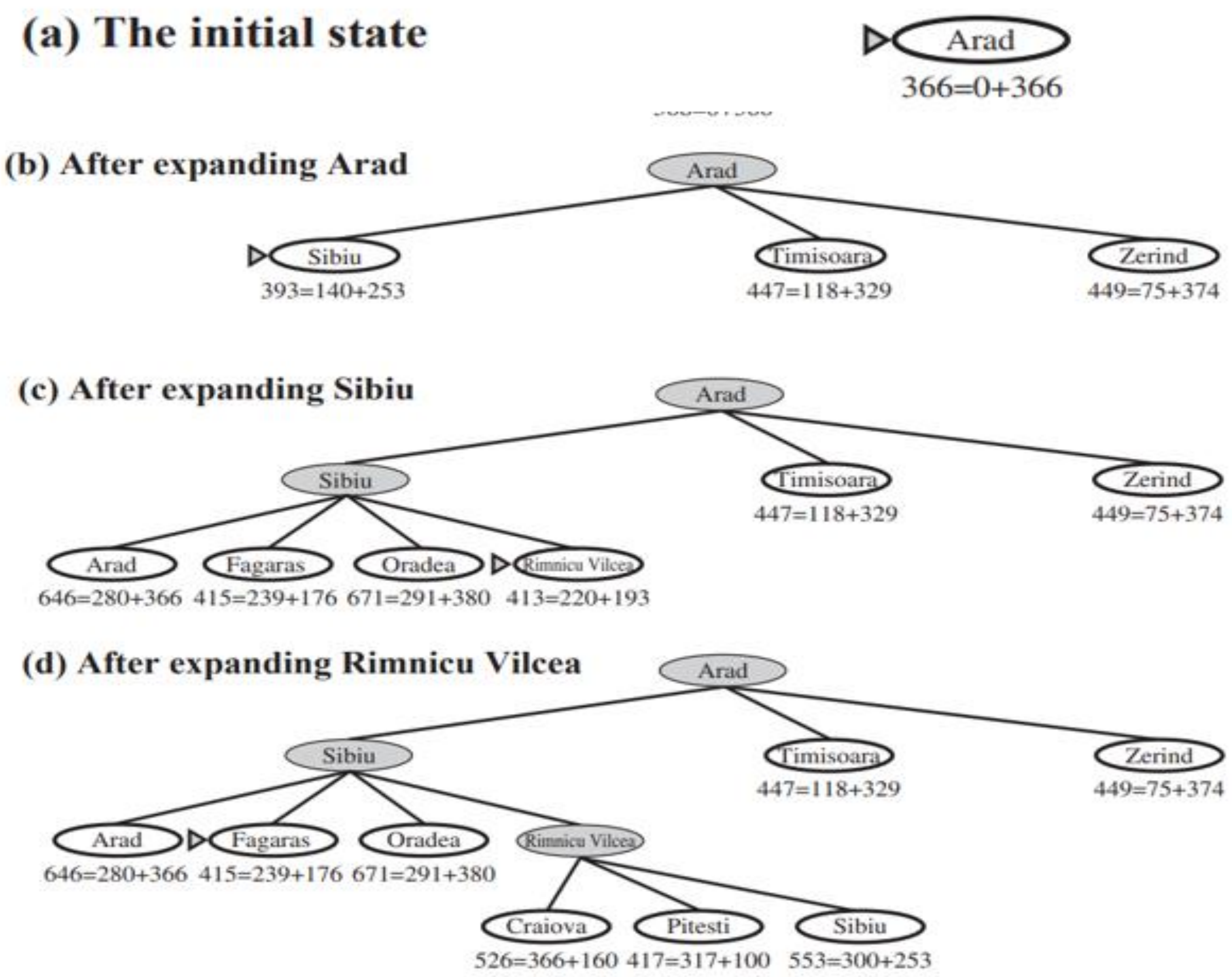
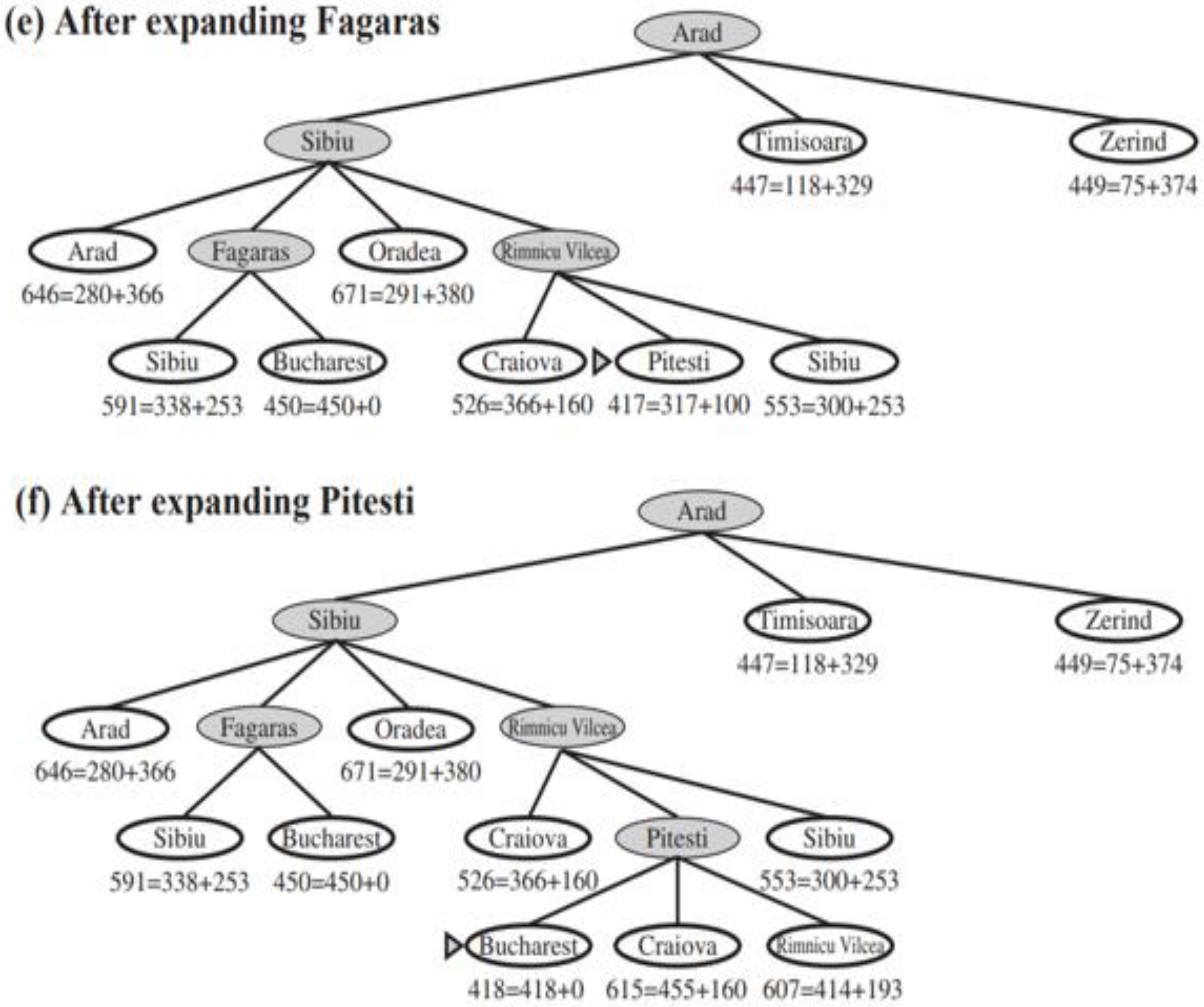
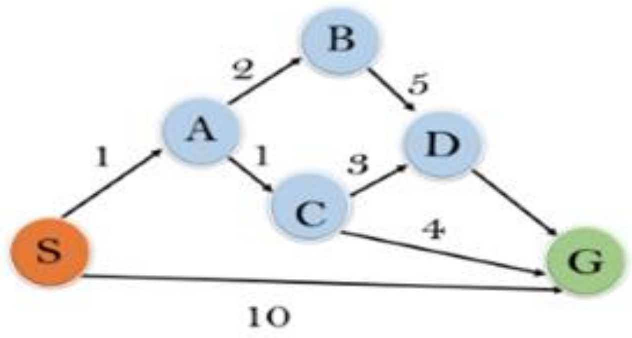


Figure below illustrates the Stages in **an A∗ search for Bucharest**. Nodes are labeled with f = g + h. The h values are the straight-line distances to Bucharest



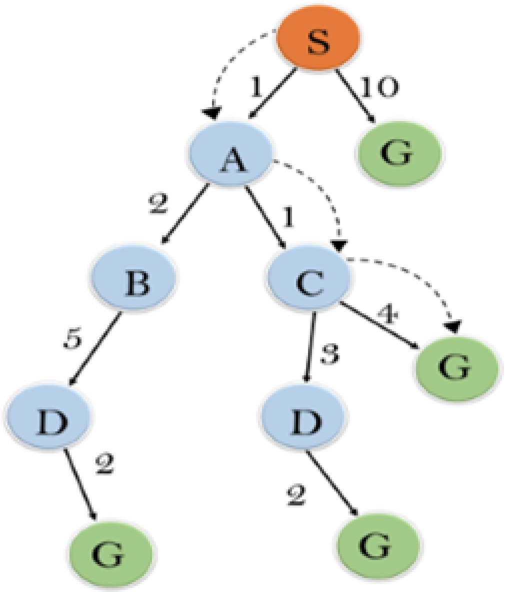
Example 2

In this example, we will traverse the given graph using the A* algorithm. The heuristic value of all states is given in the below table so we will calculate the f(n) of each state using the formula f(n)= g(n) + h(n), where g(n) is the cost to reach any node from start state. Here we will use OPEN and CLOSED list**.**



State	h(n)
S	5
A	3
B	4
C	2
D	6
G	0

Example 2 : Solution



Initialization: {(S, 5)}

Iteration1: {(S--> A, 4), (S-->G, 10)}

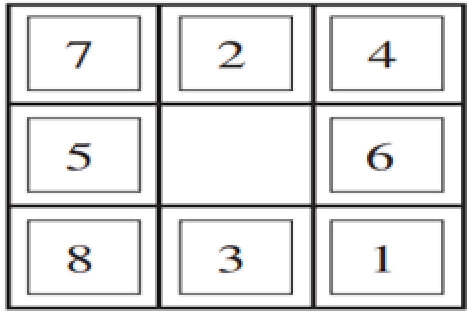
Iteration2: {(S--> A-->C, 4), (S--> A-->B, 7), (S-->G, 10)}

Iteration3: {(S--> A-->C--->G, 6), (S--> A-->C--->D, 11), (S--> A-->B, 7), (S-->G, 10)}

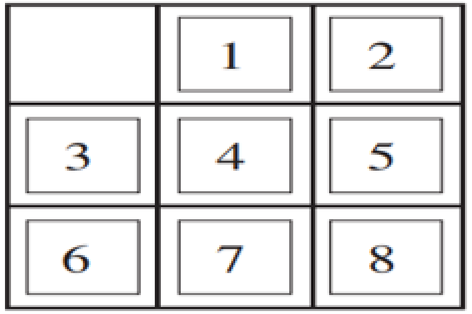
Iteration 4 will give the final result, as **S--->A--->C--->G** it provides the optimal path with cost 6.

2.1.c Heuristics Functions

- Heuristic Functions h(n) guide search algorithms by estimating the cost or distance to a goal state from the current state(n).
- Consider the 8-puzzle game. The object of the 8 puzzle is to slide the title horizontally or vertically into the empty space until the configuration matches the goal configuration.



Start State



Goal State

- Figure illustrates the A typical instance of the 8-puzzle. The solution is 26 steps long.
- The average solution cost for a randomly generated 8-puzzle instance is about 22 steps. The branching factor is about 3. (When the empty tile is in the middle, four moves are possible; when it is in a corner, two; and when it is along an edge, three.) This means that an exhaustive tree search to depth 22 would look at about $3^{22} \approx 3.1 \times 10^{10}$ states.

2.1.c Heuristics Functions

The two commonly used candidates for 8 puzzles are as follows:

h1 = the number of misplaced tiles. For Figure, all of the eight tiles are out of position, so the start state would have h1 = 8. h1 is an admissible heuristic because it is clear that any tile that is out of place must be moved at least once

h2 = the sum of the distances of the tiles from their goal positions. Because tiles cannot move along diagonals, the distance we will count is the sum of the horizontal and vertical distances. This is sometimes called the city block distance or Manhattan distance. h2 is also admissible because all any move can do is move one tile one step closer to the goal. Tiles 1 to 8 in the start state give a Manhattan distance of h2 = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18 .

A Study on heuristic functions

1. **The effect of heuristic accuracy on performance**: Experimentally it is determined that h_2 is better than h_1. That is for any node n, $h_2(n) \geq h_1(n)$.This implies that h_2 dominate h_1. Domination translates directly into efficiency. A^* using h_2 will never expand more nodes than A^* using h_1.
2. **Generating admissible heuristics from relaxed problems**: A problem with fewer restrictions on the actions is called a relaxed problem. The state-space graph of the relaxed problem is a super graph of the original state space because the removal of restrictions creates added edges in the graph.

For example, if the 8-puzzle actions are described as

- A tile can move from square A to square B if
- A is horizontally or vertically adjacent to B and B is blank,

we can generate three relaxed problems by removing one or both of the conditions:

a) A tile can move from square A to square B if A is adjacent to B.
b) A tile can move from square A to square B if B is blank.
c) A tile can move from square A to square B.

If a collection of admissible heuristics h1 . . . hm is available for a problem and none of them dominates any of the others, which should we choose? As it turns out, we need not make a choice. We can have the best of all worlds, by defining

h(n) = max{h1(n), . . . , hm(n)}

Generating admissible heuristics from subproblems: Pattern databases:

Admissible heuristics can also be derived from the solution cost of a subproblem of a given problem. For example, Figure below shows a subproblem of the 8-puzzle instance.

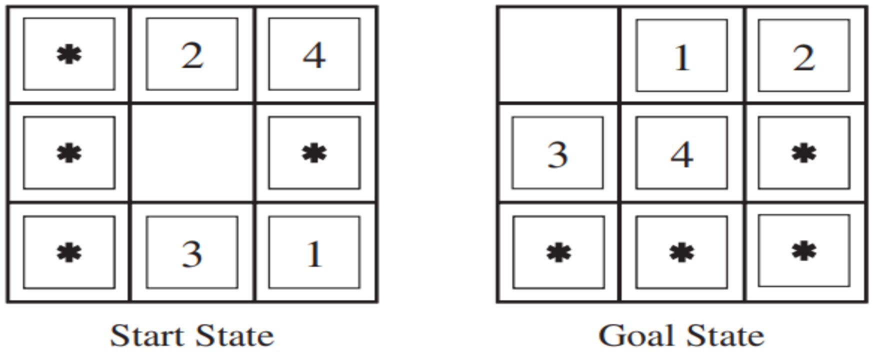


Fig: A subproblem of the 8-puzzle instance. The task is to get tiles 1, 2, 3, and 4 into their correct positions, without worrying about what happens to the other tiles.

Learning heuristics from experience

- A heuristic function, denoted as h(n), aims to approximate the solution cost starting from the state represented by node **n**. One of the strategies is to learning from practical experiences. In this context, "experience" refers to solving numerous instances of problems like 8-puzzles.
- For instance, a feature like "**number of misplaced tiles**" (**$x_1(n)$**) can be useful in predicting the distance of a state from the goal in an 8-puzzle. By gathering statistics from randomly generated 8-puzzle configurations and their actual solution costs, one can use these features to predict h(n).

Multiple features, such as **$x_2(n)$** representing the "number of pairs of adjacent tiles that are not adjacent in the goal state," can be combined using a linear combination approach:

$$\mathbf{h(n) = c_1x_1(n) + c_2x_2(n).}$$

Machine Learning

Dr. Thyagaraju G S

Machine Learning Chapters

1. Introduction to Machine Learning
2. Understanding Data
3. Basics of Learning Theory
4. Similarity Based Learning
5. Regression Analytics
6. Decision Tree Learning
7. Artificial Neural Networks
8. Clustering Algorithms

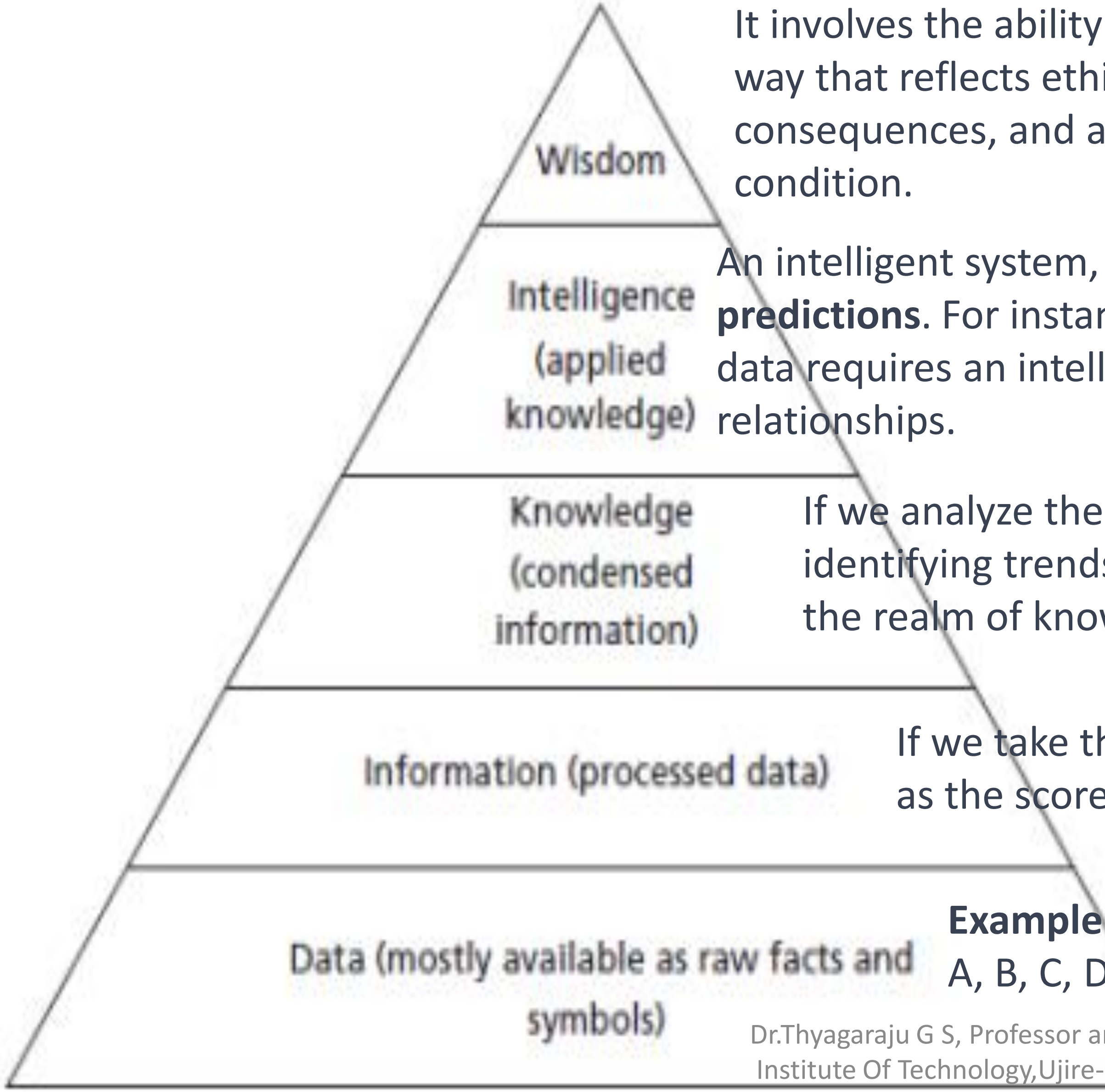
Module 2: Introduction to Machine Learning

1. Need for Machine Learning
2. Machine Learning Explained
3. Machine Learning in Relation to other Fields
   1. Machine Learning and Artificial Intelligence
   2. Machine Learning, Data Science, Data Mining and Data Analytics
   3. Machine Learning and Statistics
4. Types of Machine Learning
   1. Supervised Learning
   2. Unsupervised Learning
   3. Semi- Super Vised Learning
   4. Reinforcement Learning
5. Challenges of Machine Learning
6. Machine Learning Process
7. Machine Learning Applications

Need for Machine Learning

1. Business Organization have numerous data
2. To analyze and extract the knowledge from the data stored in the various archives of the organization
   1. To facilitate decision making
   2. Useful for design new products
   3. Improve business processes and
   4. To develop decision support system

Knowledge Pyramid



It involves the ability to apply knowledge and intelligence in a way that reflects ethical considerations, long-term consequences, and a broader understanding of the human condition.

An intelligent system, **can analyze vast datasets, learn patterns, and make predictions**. For instance, predicting **future student performance** based on historical data requires an intelligent system that can discern underlying trends and relationships.

If we analyze the test scores (information) and draw conclusions, such as identifying trends, strengths, weaknesses, or correlations, we are moving into the realm of knowledge

If we take the numbers from the previous example (1, 5, 7, 3) and label them as the scores of students in a test, we now have information

Example: A list of numbers (e.g., 1, 5, 7, 3) or a series of characters (e.g., A, B, C, D) would be considered data.

Popularity of Machine Learning

1. High Volume of Available data to manage
2. The Cost of storage has reduced
3. Availability of Complex Algorithms

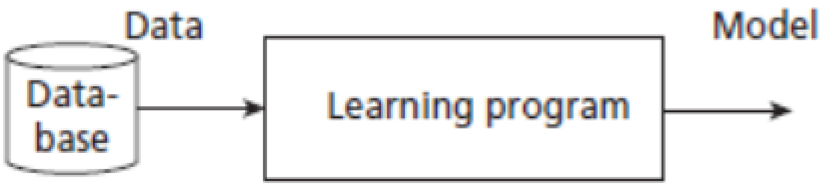
What is Machine Learning ?

- **Machine Learning is the field of study that gives the computers ability to learn without being programmed**.[Arthur Samuel]
  - Here the input Data is used to **develop intelligent models**. The models will be used to predict new inputs.
  - The aim of ML is **to learn a model or set of rules** from the given data set automatically so that it can predict the unknown data correctly.

Learning systems

For Humans

For Machine Learning



What is a Model?

A Model can be any one of the following:

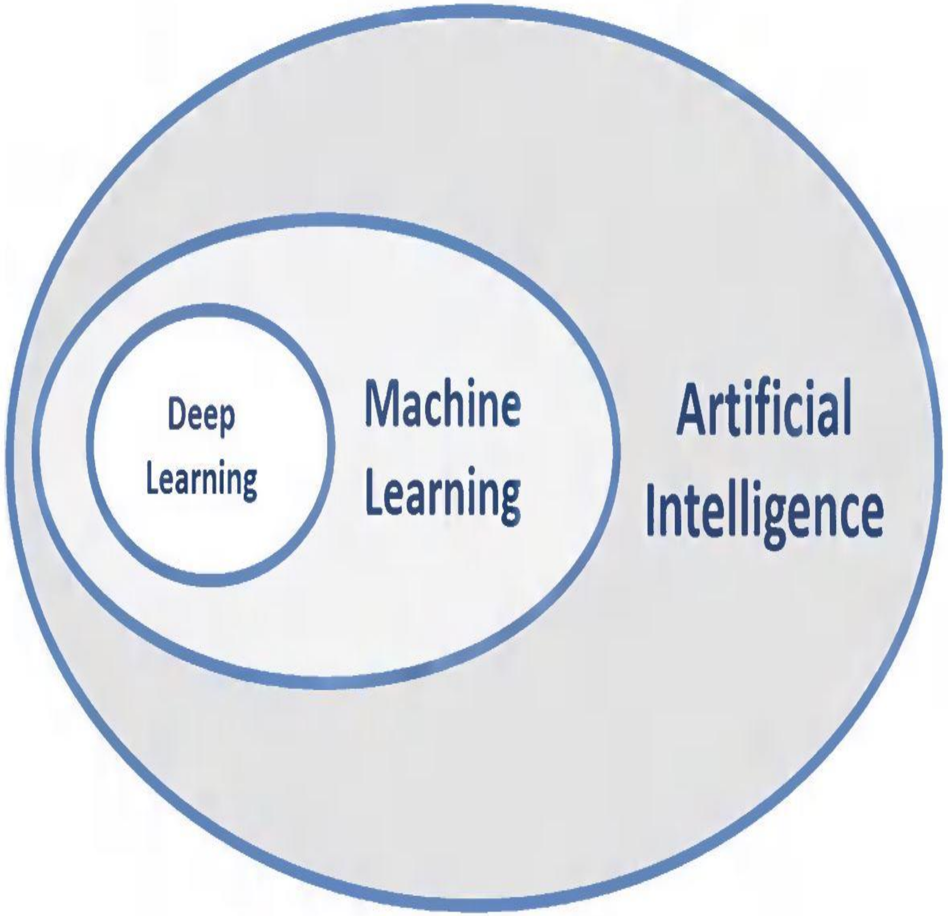
1. Mathematical Equation
2. Relational diagrams like Graphs/Trees
3. Logical if/else rules
4. Groupings called clusters

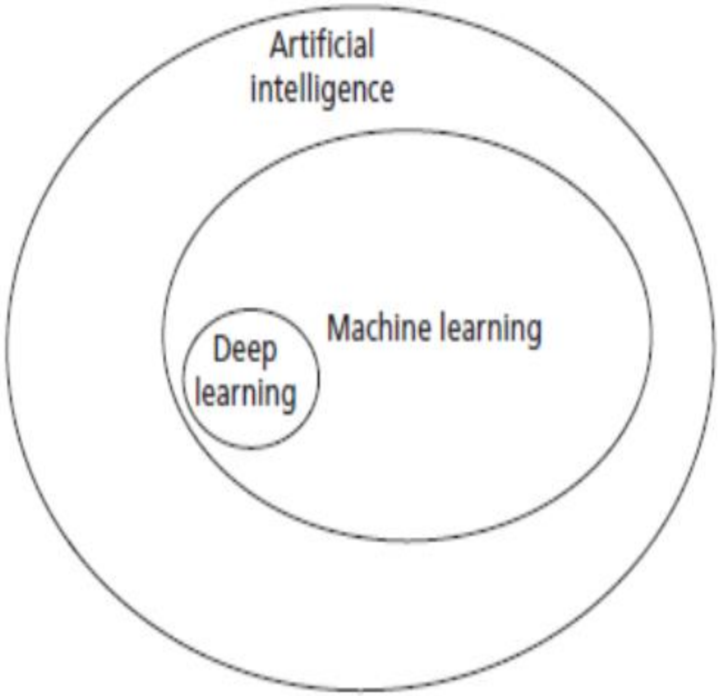
Tom Mitchell definition of Machine learning

- Tom Mitchell, a computer scientist and professor at Carnegie Mellon University, provided a widely cited and influential definition of machine learning in his book titled "Machine Learning" (1997). The definition is as follows:
- "***A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E***."

Relationship of ML with Other Fields

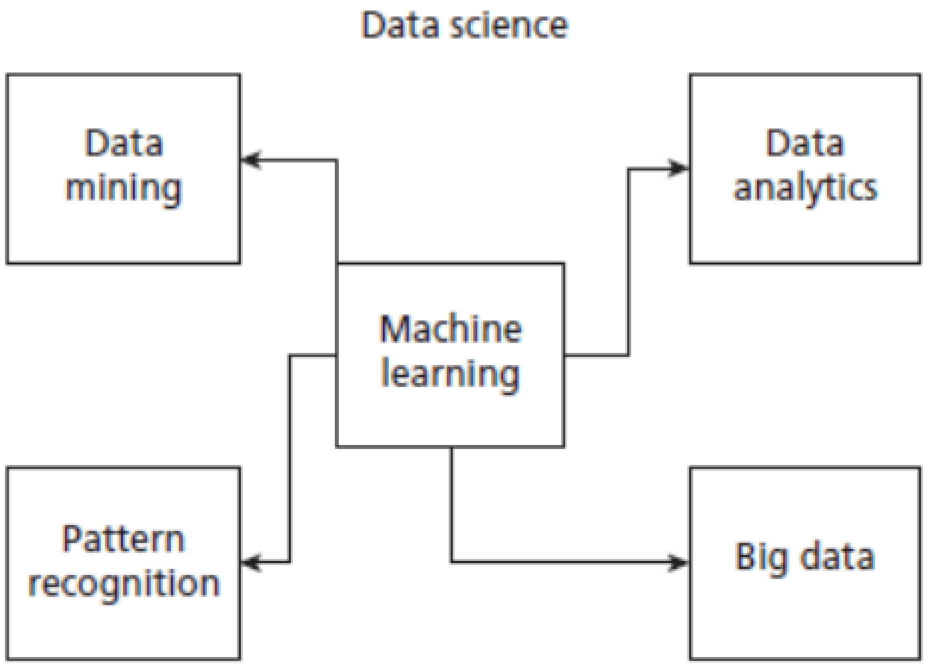
Relationship of AI with ML





Machine Learning and Data Science

Data science is an “umbrella term” covering from data collection to data analysis.



CHARACTERISTICS OF BIG DATA

1. **Volume:** Big Data involves massive amounts of data that can be in petabytes or exabytes.
2. **Velocity:** Describes the speed at which data is generated, collected, and processed.
3. **Variety:** Encompasses the different types of data sources and formats. Big Data often includes structured data (e.g., databases), unstructured data (e.g., text, images, videos), and semi-structured data (e.g., JSON or XML files).
4. **Veracity:** Refers to the quality and reliability of the data. Big Data sources can be messy and may include inaccuracies, inconsistencies, and errors.
5. **Value:** Focuses on the importance of turning data into value. The ultimate goal of working with Big Data is to extract meaningful insights and value from the massive amounts of data collected.

Data Science and Data Mining

- **Data Science** is a multidisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data.
- **Data Mining** is a specific step within the broader field of data science. It involves the process of discovering patterns, relationships, and insights from large datasets using various techniques, including statistical analysis, machine learning, and artificial intelligence.

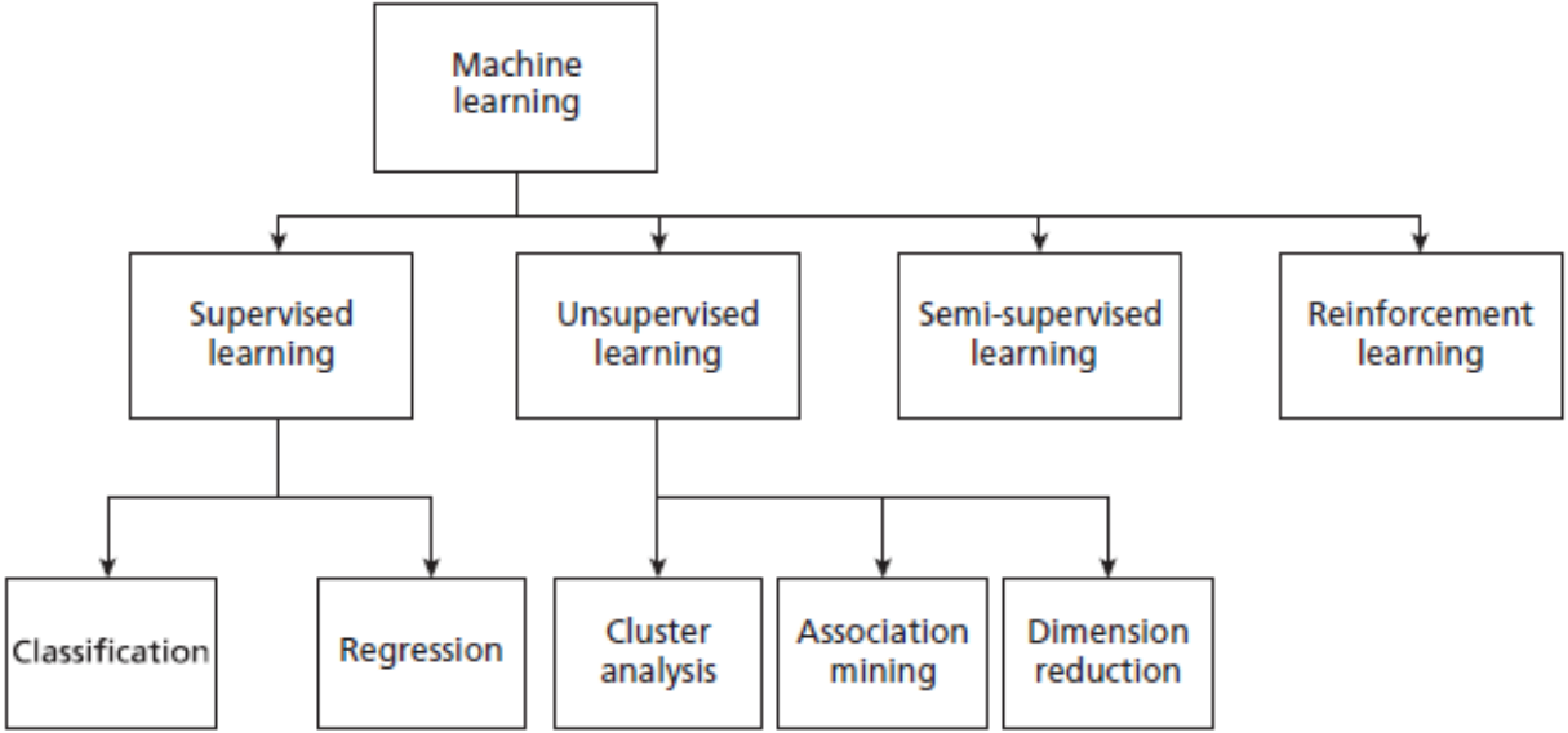
Data science and Data analytics / Pattern recognition

- **Data Science, Data Analytics, and Pattern Recognition** are closely related fields, and they share common goals of extracting insights and knowledge from data. However, each field has its own focus and methodologies.
- **Data Science is a multidisciplinary field** that involves the use of scientific methods, processes, algorithms, and systems to extract insights and knowledge from structured and unstructured data.
- **Data Analytics is the process** of examining, cleaning, transforming, and modeling data to discover useful information, draw conclusions, and support decision-making.
- **Pattern Recognition** is the process of automatically recognizing patterns in data and making decisions based on those patterns.

Machine Learning and Statistics

- **Machine Learning** is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn from data and make predictions or decisions without explicit programming.
- **Statistics** is a branch of mathematics that involves collecting, analyzing, interpreting, presenting, and organizing data. It provides methods for making inferences from data, drawing conclusions, and quantifying uncertainty.

Machine Learning types



Dr.Thyagaraju G S, Professor and HoD, Department of CSE, SDM Institute Of Technology,Ujire-574240. Source Book : S. Sridhar, M Vijayalakshmi “Machine Learning”. Oxford, 2021

Labelled and Unlabeled Data

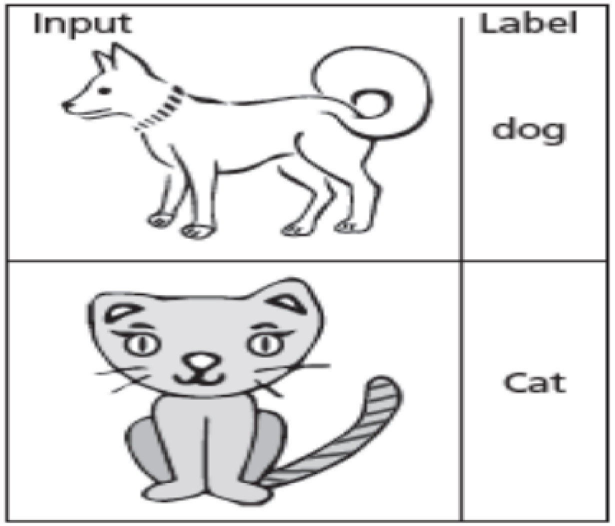
- Data is a raw fact.
- Data is represented in the form of data table
- Data also can be referred to as a data point, sample or an example
- Features are attributes or characteristics of an object.Columns of table are attributes.
- The most important attribute is LABEL. Label is the feature that we aim to predict.
- There are two types of data : **Labelled Data and Unlabeled Data**

Labelled Data

Table 1.1: Iris Flower Dataset

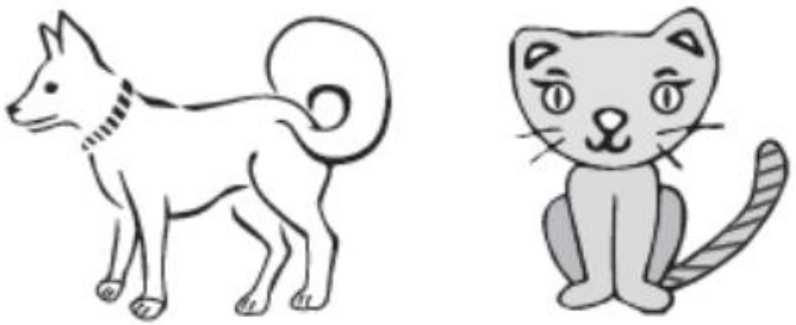
S.No.	Length of Petal	Width of Petal	Length of Sepal	Width of Sepal	Class
1.	5.5	4.2	1.4	0.2	Setosa
2.	7	3.2	4.7	1.4	Versicolor
3.	7.3	2.9	6.3	1.8	Virginica

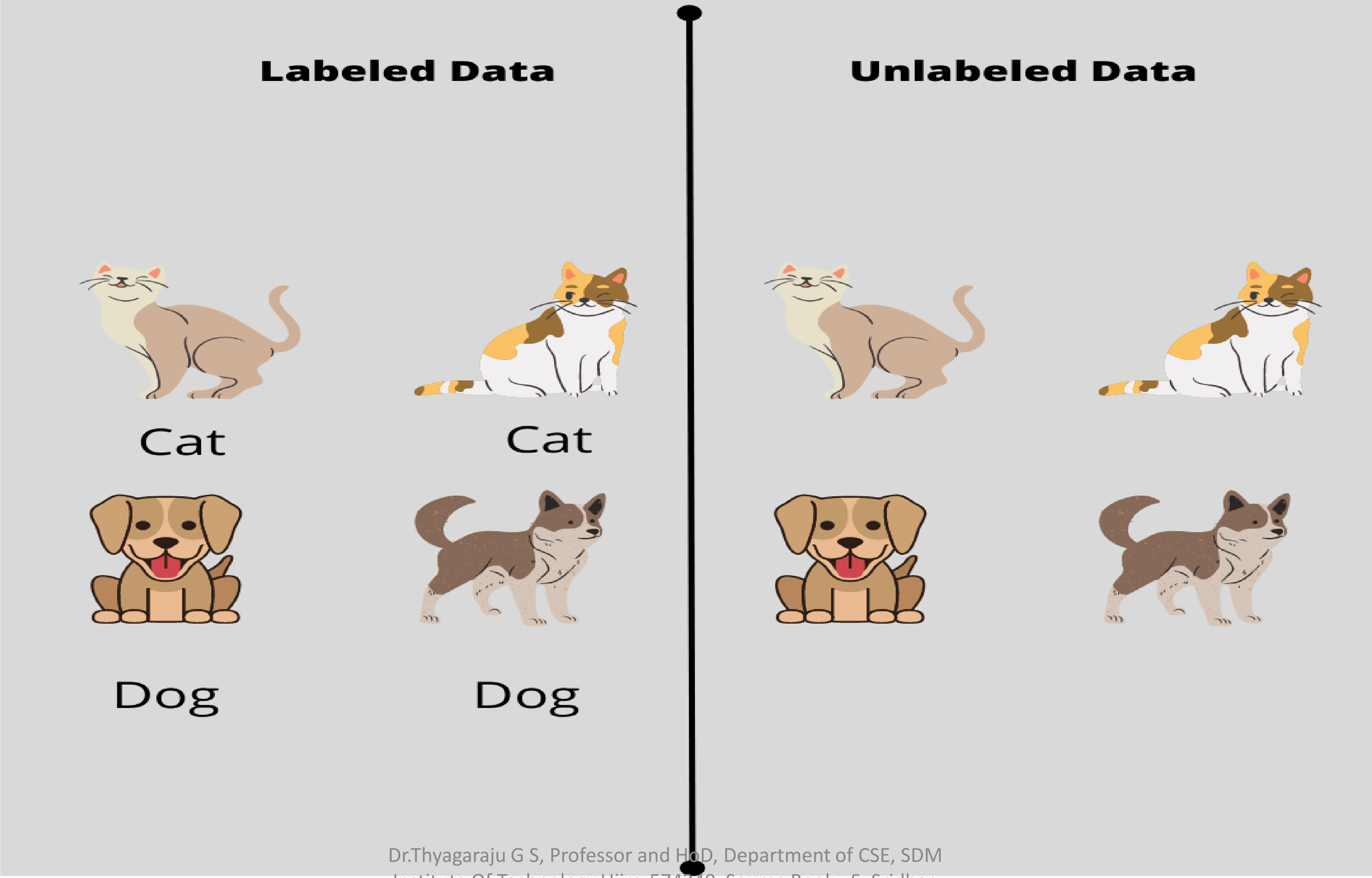
A dataset need not be always numbers. It can be images or video frames. Deep neural networks can handle images with labels. In the following Figure 1.6, the deep neural network takes images of dogs and cats with labels for classification.



Unlabelled Data

DATA THAT IS NOT ASSOCIATED WITH LABELS ARE CALLED UNLABELLED DATA





Labeled Data
Unlabeled Data
Cat
Cat
Dog
Dog

Supervised Learning

- **Supervised learning** is a type of machine learning where the algorithm is trained on a labeled dataset, meaning that the input data used for training is paired with corresponding output labels. The goal of supervised learning is to learn a mapping from input features to the target output by generalizing patterns from the labeled training data. Once trained, the model can make predictions or classifications on new, unseen data.

Key Components of SL

- **Input Features (X)**
- **Output Labels (Y)**
- **Training Data**
- **Model**
- **Testing Data**

Process of Supervised Learning

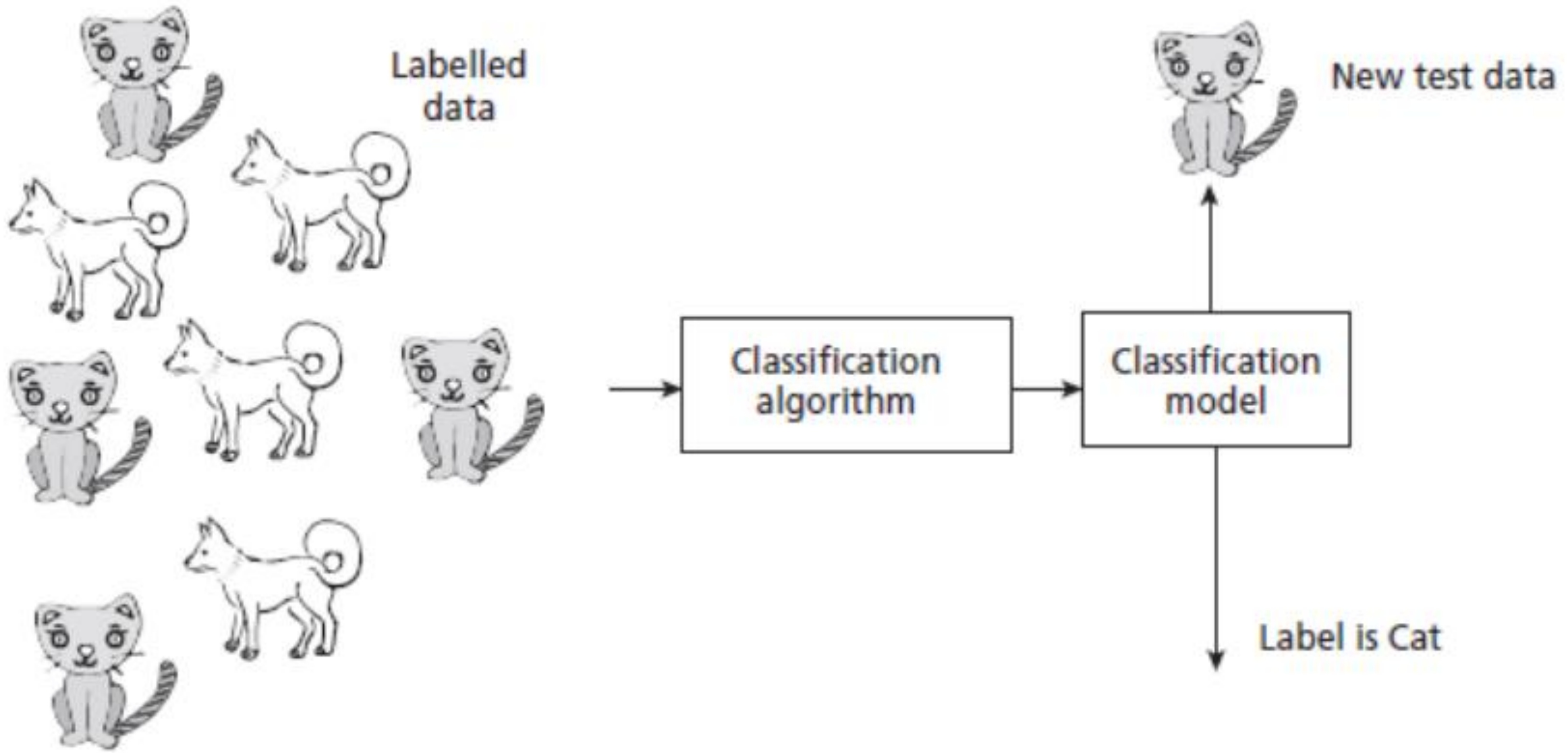
1. **Data Collection:** Gather a labeled dataset that includes input features and corresponding output labels.
2. **Data Preprocessing:** Clean and preprocess the data, handling missing values, scaling features, and encoding categorical variables.
3. **Model Selection:** Choose an appropriate supervised learning algorithm based on the problem type (regression or classification) and characteristics of the data.
4. **Training:** Feed the labeled data into the chosen model, allowing it to learn the patterns and relationships between input features and output labels.
5. **Evaluation:** Assess the performance of the trained model on a separate validation or test dataset using metrics relevant to the specific task (e.g., mean squared error for regression or accuracy for classification).
6. **Prediction:** Deploy the trained model to make predictions or classifications on new, unseen data.

Types of Supervised Learning:

1. **Classification:** In classification tasks, the goal is to assign input data to specific categories or classes. Examples include spam detection, image recognition, or sentiment analysis.

2. **Regression:** In regression tasks, the goal is to predict a **continuous output variable.** Examples include predicting house prices, temperature, or stock prices.

1.Classification



Algorithms : Classification

	Name	Type	Application	Strengths
1	Logistic Regression	Linear Model	Binary classification problems	Simple, interpretable, and effective for linearly separable data
2	Decision Trees	Tree based Model	Both binary and multi-class classification problems	Intuitive, handles non-linear relationships, and can automatically handle feature interactions
3	**Random Forest:**	Ensemble Model (multiple decision trees)	Classification and regression problems	Robust, handles overfitting, and often provides high accuracy.
4	**Support Vector Machines (SVM)**	Linear and Non-linear	Binary and multi-class classification problems	Effective in high-dimensional spaces, versatile through the use of different kernels.
5	**K-Nearest Neighbors (KNN)**	Instance-based Model	Classification and regression problems	Simple, non-parametric, and effective for small datasets
6	**Naive Bayes:**	Probabilistic Model	Text classification, spam filtering	Simple, efficient, and performs well in high-dimensional spaces.
7	**Neural Networks (Deep Learning)**	Neural Network Model	Image recognition, natural language processing, complex patterns	Capable of learning intricate patterns, suitable for large datasets.

2. Regression

- Regression algorithms are used in machine learning to model and predict continuous numeric values.

2. Regression

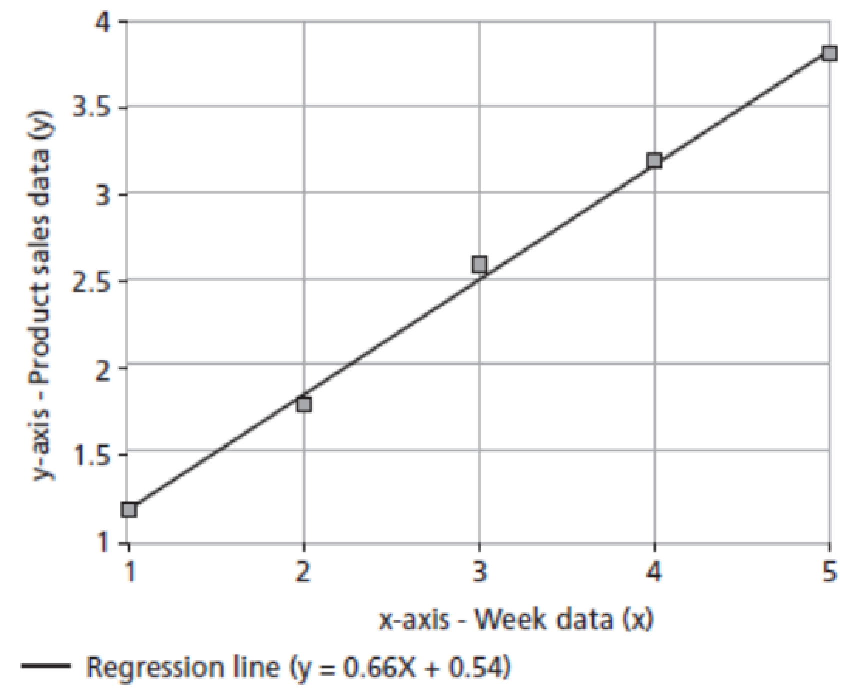
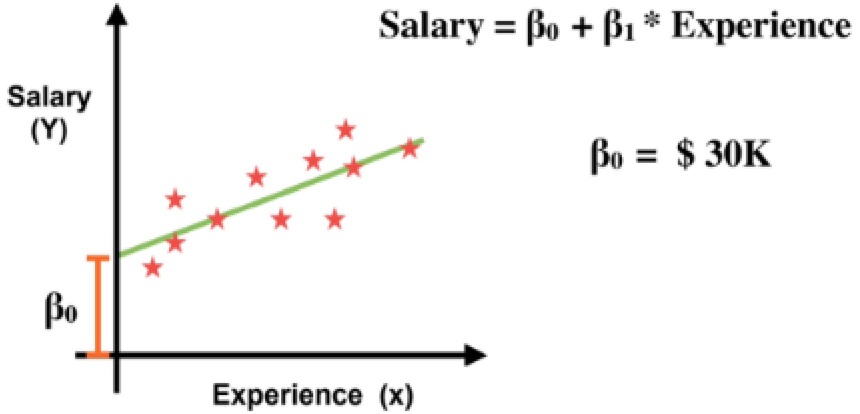
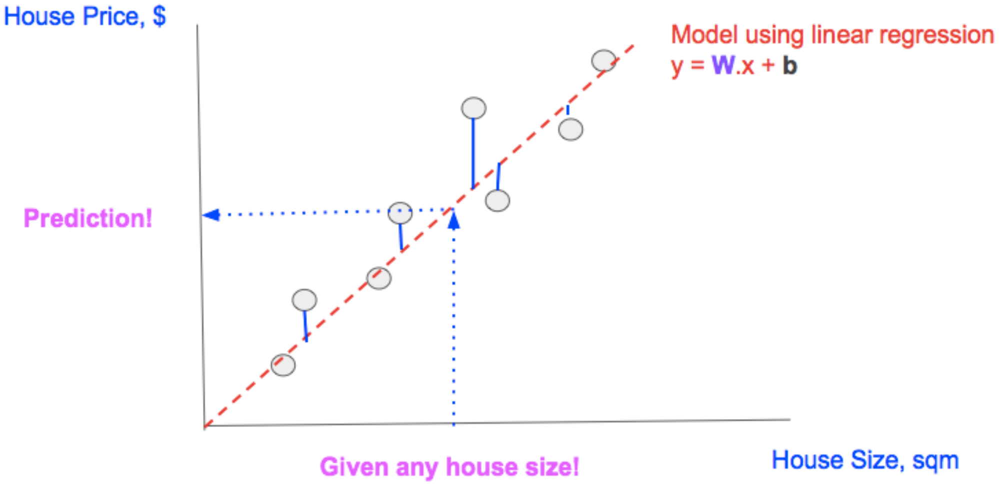


Figure 1.8: A Regression Model of the Form $y = ax + b$

$$\text{product sales} = 0.66 \times \text{Week} + 0.54$$





Algorithms : Regression

	Name	Type	Application	Strengths
1	Linear Regression	Linear Model	Predicting a continuous outcome based on one or more input features.	Simple, interpretable, and effective for linear relationships
2	**Decision Trees**	Tree-based Model	Regression tasks where the relationship between features and target is non-linear.	Can capture complex relationships and interactions in the data.
3	**Random Forest**	Ensemble Model (multiple decision trees)	Regression tasks where high accuracy and robustness are required.	Reduces overfitting, provides high accuracy, and handles non-linearity.
4	**Support Vector Regression (SVR)**	Support Vector Machine for Regression	Predicting a continuous outcome using a hyperplane with a margin of tolerance.	Effective in high-dimensional spaces and capable of handling non-linear
5	**K-Nearest Neighbours (KNN) Regression**	Instance-based Model	Regression tasks where the output is influenced by the values of nearby data points.	Simple, non-parametric, and suitable for small to medium-sized datasets

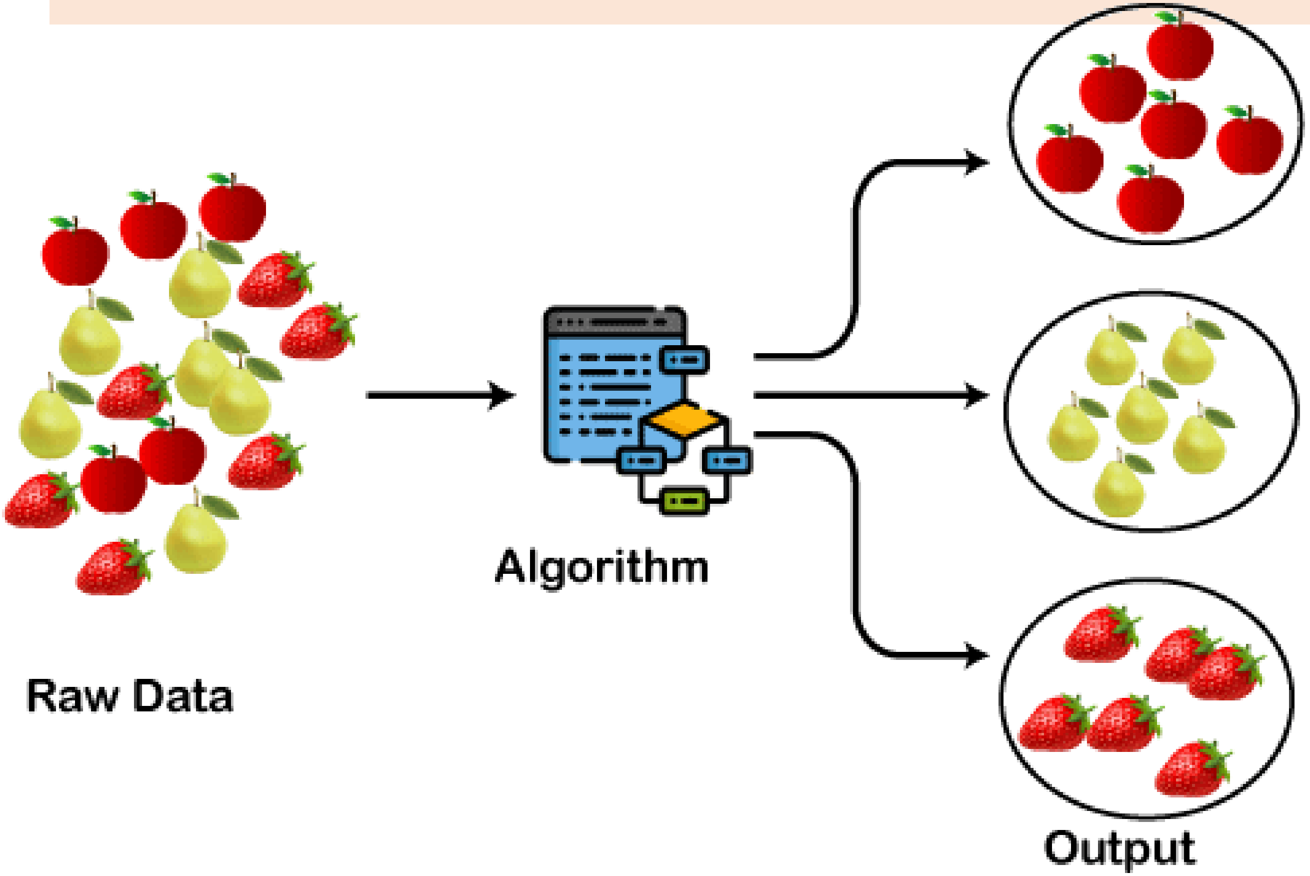
Unsupervised Learning

- **Unsupervised learning** is a type of machine learning where the algorithm is given data without explicit instructions on what to do with it. The system tries to learn the patterns and the structure from the data without labeled responses to guide the learning process. Unsupervised learning is often used for exploratory data analysis and pattern discovery. The two main types of unsupervised learning are clustering and dimensionality reduction.
- **Unsupervised learning** is particularly useful in scenarios where labeled data is scarce or unavailable. It allows the algorithm to discover hidden patterns and structures in the data without explicit guidance. The choice of algorithm depends on the specific task, characteristics of the data, and the goals of the analysis.

Types of Unsupervised Learning

1. Clustering
2. Dimensionality Reduction
3. Association Rule Mining

Clustering



Clustering

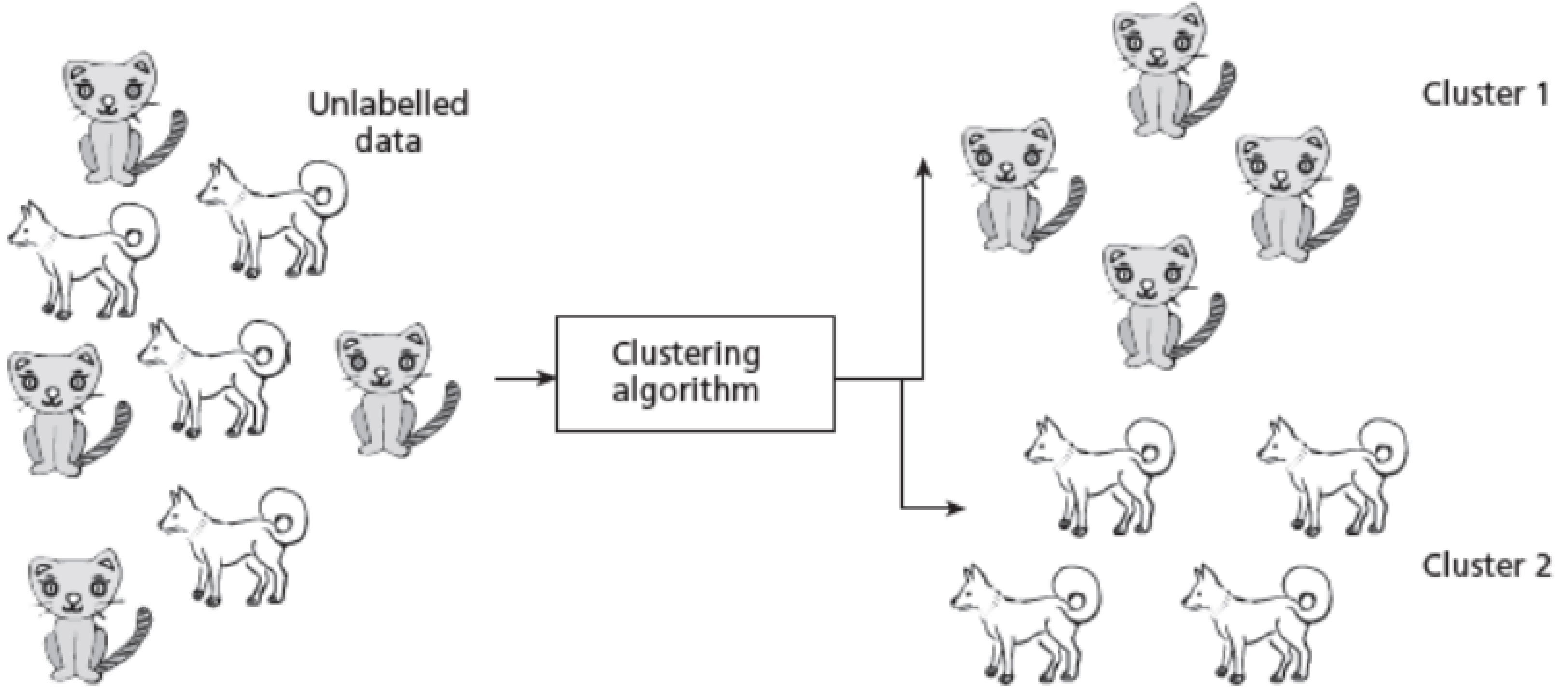


Figure 1.9: An Example Clustering Scheme

Algorithms : Clustering

	Name	Type	Application	Strengths
1	**K-Means Clustering**	Partitioning Method	Grouping data into k clusters based on similarity.	Simple, efficient, and effective for well-separated clusters.
2	**Hierarchical Clustering**	Agglomerative or Divisive	Creating a tree of clusters, revealing relationships between data points.	Visual representation of cluster hierarchy, useful for small to medium-sized datasets.

Algorithms : Dimensionality Reduction

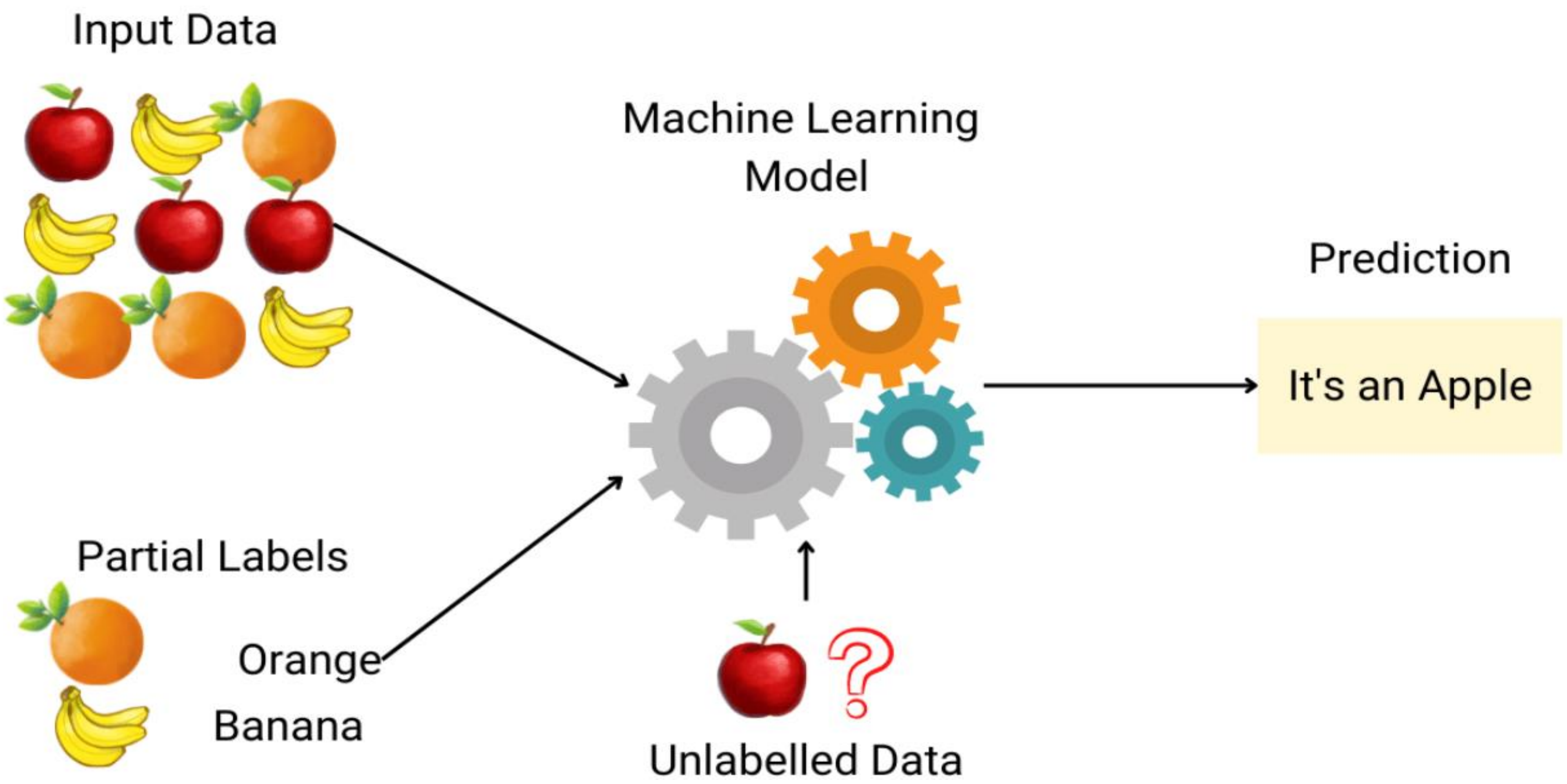
	Name	Type	Application	Strengths
1	**Principal Component Analysis (PCA)**	Linear Transformation	Reducing the dimensionality of data while retaining most of the variance.	Widely used for feature extraction and visualization.
2	**Autoencoders**	Neural Network-based	Learning a compressed, lower-dimensional representation of the input data.	Can capture complex non-linear relationships in the data.
3	**Independent Component Analysis (ICA)**	Linear Transformation	Separating a multivariate signal into additive, independent components.	Useful for blind source separation and feature extraction.

Supervised learning	Unsupervised learning
Input data is labelled	Input data is unlabelled
There is a training phase	There is no training phase
Data is modelled based on training dataset	Uses properties of given data for classification
Divided into two types: Classification and Regression	Most popular types: Clustering and Dimensionality reduction
Known number of classes (for classification)	Unknown number of classes

Semi **Supervised** Learning

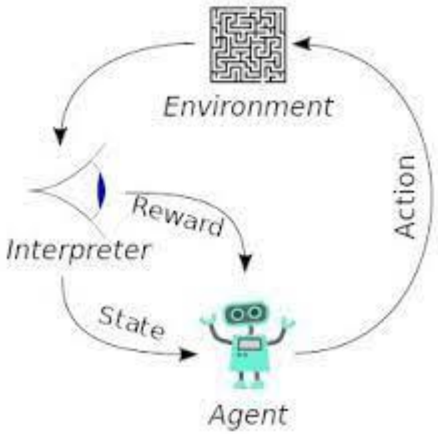
- Semi-supervised learning is a type of machine learning where the algorithm is trained on a dataset that contains both labeled and unlabeled data. In other words, only a subset of the training data has explicit labels, while the majority of the data is unlabeled. Semi-supervised learning aims to leverage the information from both labeled and unlabeled data to improve the performance of the model.

Semi Supervised Learning

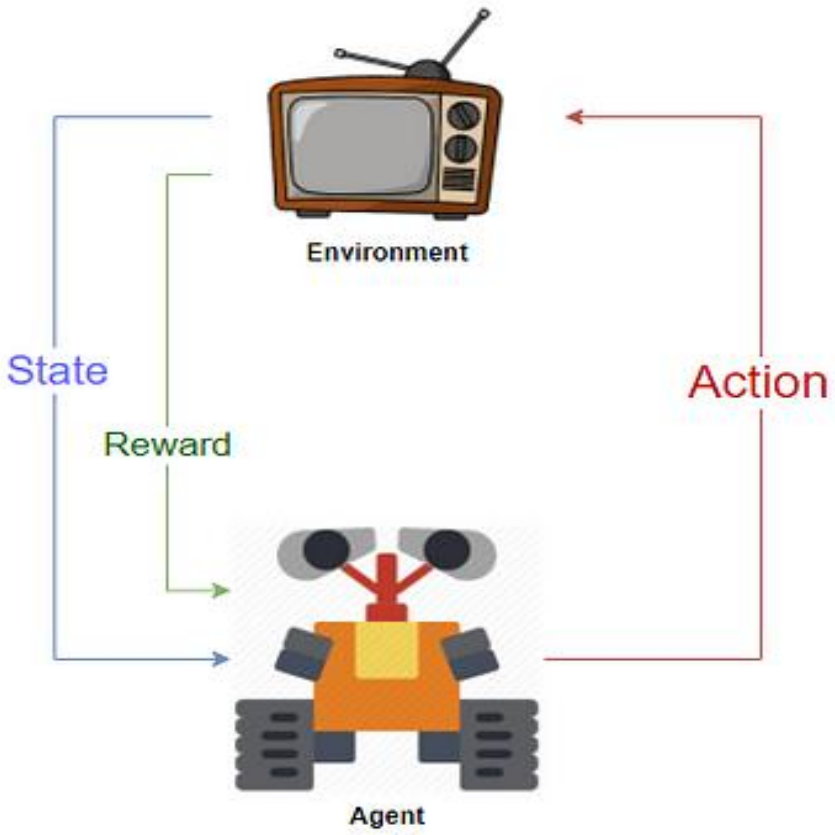


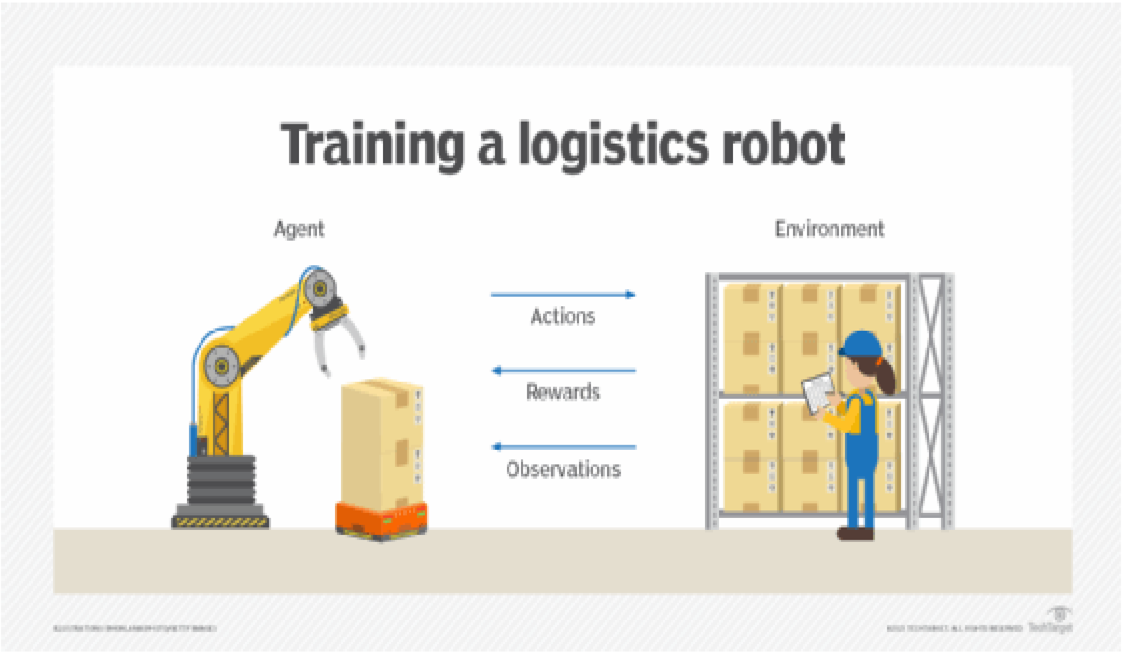
Reinforcement Learning

- Reinforcement Learning (RL) is a type of machine learning paradigm where an agent learns to make decisions by interacting with an environment. The agent takes actions, receives feedback in the form of rewards or penalties, and learns to optimize its behavior over time to achieve a specific goal. In RL, the agent is not explicitly told which actions to take but discovers the optimal strategy through trial and error.



Reinforcement Learning





Reinforcement Learning

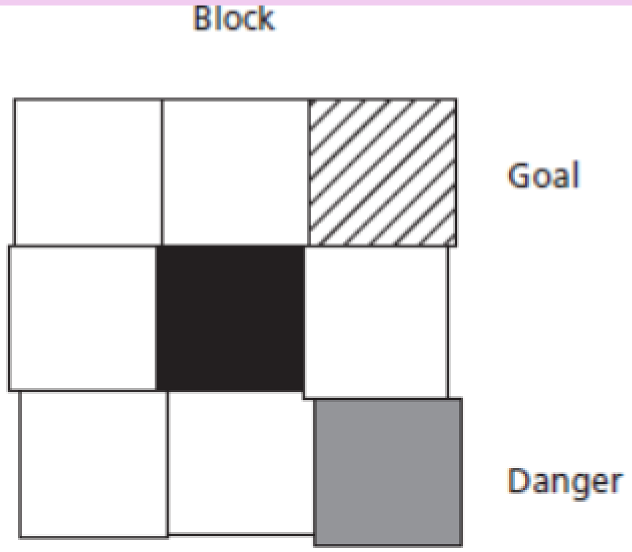
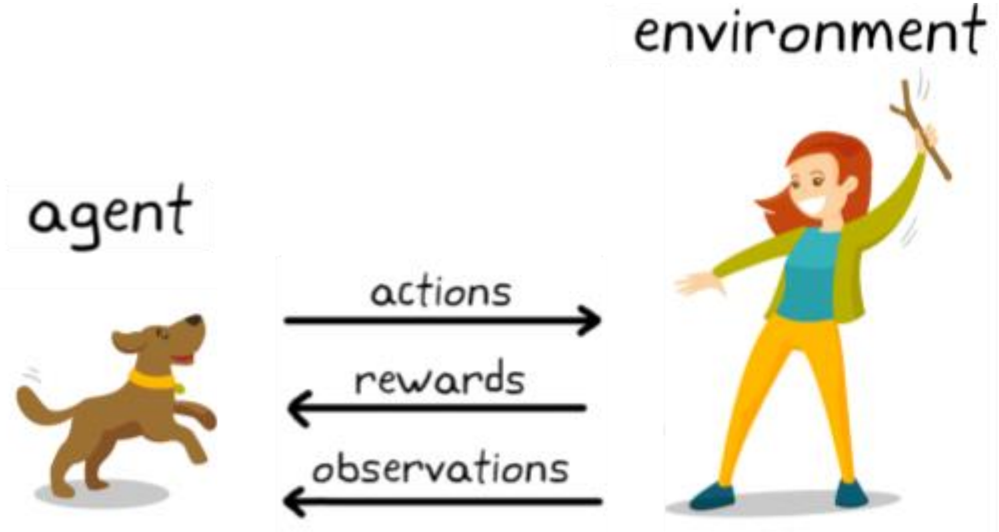
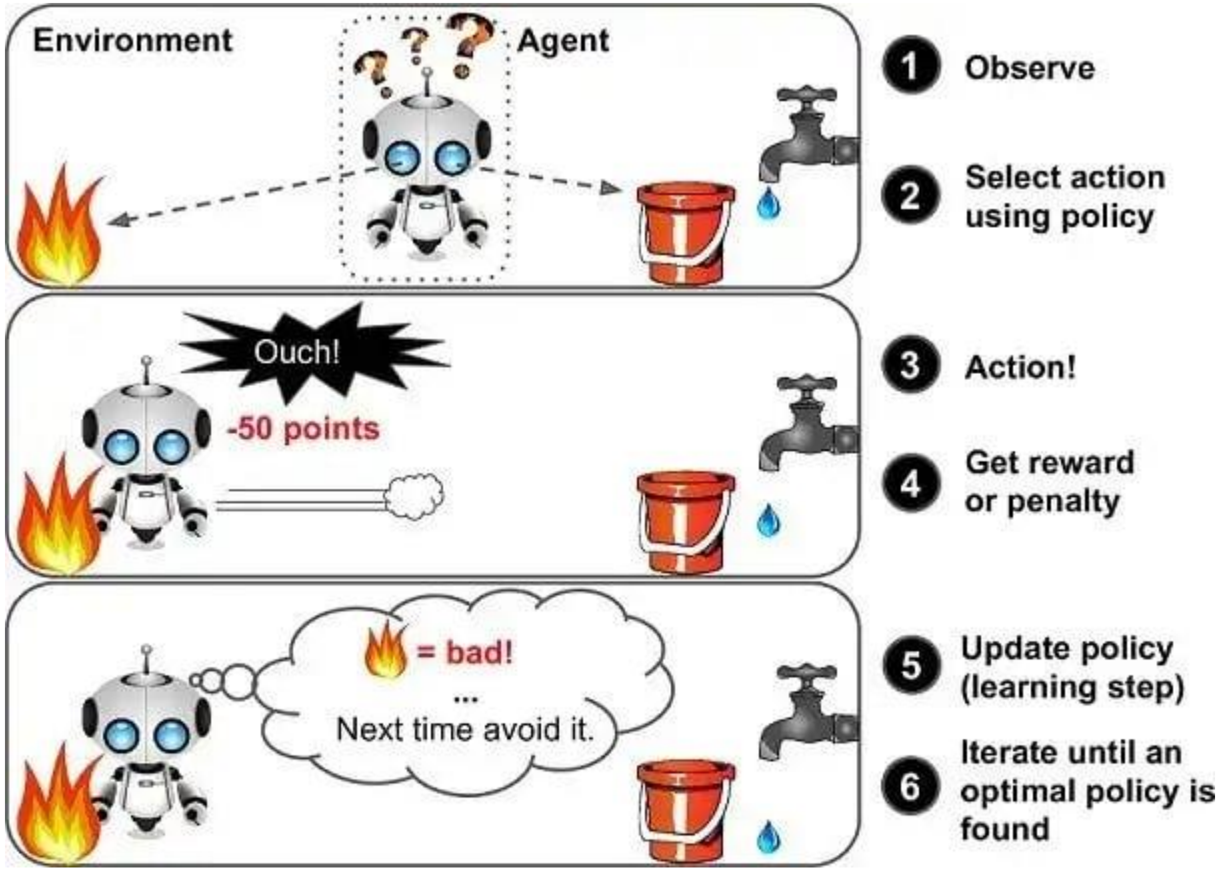


Figure 1.10: A Grid game



Reinforcement Learning



Challenges of Machine Learning

1. ILL-POSED PROBLEMS – PROBLEMS WHOSE SPECIFICATIONS ARE NOT CLEAR
2. HUGE DATA
3. HUGE COMPUTATION POWER
4. COMPLEXITY OF ALGORITHMS
5. BIAS-VARIANCE

Machine Learning Process

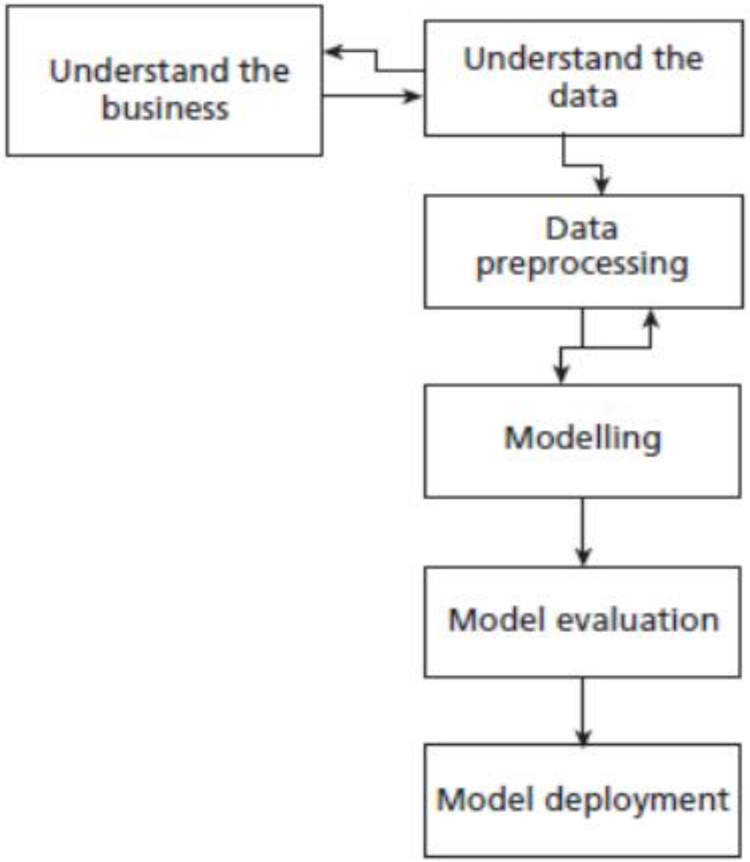


Figure 1.11: A Machine Learning/Data Mining Process

Machine Learning Applications

- Sentiment Analysis
- Recommendation Systems
- Voice Assistants
- Google Maps
- Facial Recognition
- Object Detection
- NLP
- Financial Fraud Detection
- Energy Management
- E-commerce
- Gaming

Machine Learning Applications

Sl.No	Problem Domain	Applications
1	Business	Predicting the banckruptcy of a business firm
2	Banking	Prediction of bank loan defaulters and detecting credit card frauds
3	Image Processing	Image search engines, object identification, image classification and generating synthetic images
4	Audio/ Voice	Chatbots like Alexa, Microsoft Cortana. Developing chatbots for customer support, speech to text and text to voice
5	Telecommunication	Trend analysis and identification of bogus calls, fraudulent calls and the callers.

Machine Learning Applications

Sl.No	Problem Domain	Applications
6	Games	Game programs for Chess, GO and Video games
7	Natural Language Translation	Google Translation, Text Summarization and sentiment and analysis
8	Web Analysis and Services	Identification of Access patterns, detection of e-mail spams, viruses, personalized web services, search engines like Google, detection of promotion of user websites
9	Medicine	Prediction of disease
10	Multimedia and Security	Face Recognition/Identification biometric projects like identification of a person from a large image or video database.

Machine Learning

Dr. Thyagaraju G S

2.3 Understanding Data

- Under development