REVA UNIVERSITY
Bengaluru, India

M23DE0201 – Machine Learning

II Semester MCA D Academic Year : 2024 - 2025

School of Computer Science and Applications

Dr.P SREE LAKSHMI
Assistant Professor

Unit I

Introduction to Machine Learning

Lecture – 1

Agenda:

- COURSE OBJECTIVES
- COURSE CONTENT
- READING MATERIAL
- COURSE OUTCOMES
- OVERVIEW OF THE PREREQUISITE

Course Objectives:

- Develop a deeper understanding of basic components and categories of Machine Learning
- Explore on various data pre-processing techniques and various supervised based Machine learning algorithms.
- Understand the characteristics and its limitation of various unsupervised based Machine learning algorithms
- Examine the limitations of various machine learning algorithms upon Reinforcement Learning

Course content:

UNIT 1: **[10 Hours]**

Introduction to Machine learning: Overview of ML, broad categories of Machine learning- Supervised, Unsupervised, Semi-supervised, and Reinforcement Learning, Applications areas of Machine Learning. Data Pre-processing, Training and Choosing Predictive Models, Model Evaluation and Validation of unseen data instances.

UNIT 2: **[14 Hours]**

Supervised Learning: Introduction, Classification and Linear Regression, k-Nearest Neighbor, Linear models, Decision Trees, Naive Bayes Classifiers, Support Vector Machine - Soft Margin and Non-Linear SVM classification. Neural Networks - The Perceptron, MLP and Backpropagation, Train a DNN, Construction and Execution phase, How to use the Neural Network, Fine- tuning the Hyperparameters, The Number of Hidden Layers, Activation Functions.

Visual Cortex Architecture, Convolutional Layers, Filters, Common CNN architectures, LexNet , AlexNet, GoogleNet and ResNet

Course content:

UNIT 3: **[14 Hours]**

Unsupervised Learning: Introduction, types and challenges, preprocessing and scaling of datasets, Dimensionality reduction, feature extraction. Principal Component Analysis (PCA), k-means, agglomerative and DBSCAN clustering algorithms.

UNIT 4: **[14 Hours]**

Reinforcement: Introduction, Learning How to Optimize Rewards, Policy Search, Neural Network Policies, Action Evaluation: Credit Assignment problem, Using Policy Gradients, Markov Decision Processes, Q learning – function,Using Deep QLearning to learn how to play Pacman

READING MATERIAL

Text Books:

1 Andreas C Muller & Sarah Guido , Introduction of Machine Learning with Python, O'Reilly & Shroff publishers. Chapters 1, 2 and 3.
2 Sam Bill Lubanovic, Introducing Python , Oriely Publications, 1st Edition , Chapters 1-6
3 Tom M Mitchell , Machine Learning, McGraw Hill Education publication – 2013. Chapter 13.
4 EthemAlpaydi, Introduction to Machine Learning, Second Edition, The MIT Press, 2015
5 ShaiShalev- Shwartz and Shai Ben David, Understanding Machine Learning: From Theory to Algorithms, First Edition, Cambridge University Press, 2014
6. EthemAlpaydin , Machine Learning , PHI learning private limited. Chapter 1, 7, 16, 18, 19.

READING MATERIAL

References Books:

1 Sudharsan Ravichandran , Reinforcement Learning with Python: Master reinforcement and deep reinforcement learning using OpenAI Gym and TensorFlow, Packt Publishers, 2018.

2 Bharath Ramsundar and Reza BosaghZadeh TensorFlow for Deep Learning, O‟Reilly Publications, 2018.

3 Peter Flach, Machine Learning: The Art and Science of algorithms, Cambridge University Press. Chapter 12

4 David barber, Bayesian Reasoning and Machine Learning, Cambridge University Press. Chapter 13, 15

5 Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006

Course Outcomes:

At the end of the course students will be able to:

CO1 Understand the fundamental issues and challenges of machine learning: data, model selection, model complexity etc.

CO2 Analyze the characteristics, strengths and weaknesses of Supervised, Unsupervised and Reinforcement Learning techniques and perform evaluation and model selection

CO3 Appreciate the underlying mathematical relationships within and across machine learning algorithms.

CO4 Build various machine learning models in a range of real-world applications based on suitable model parameters.

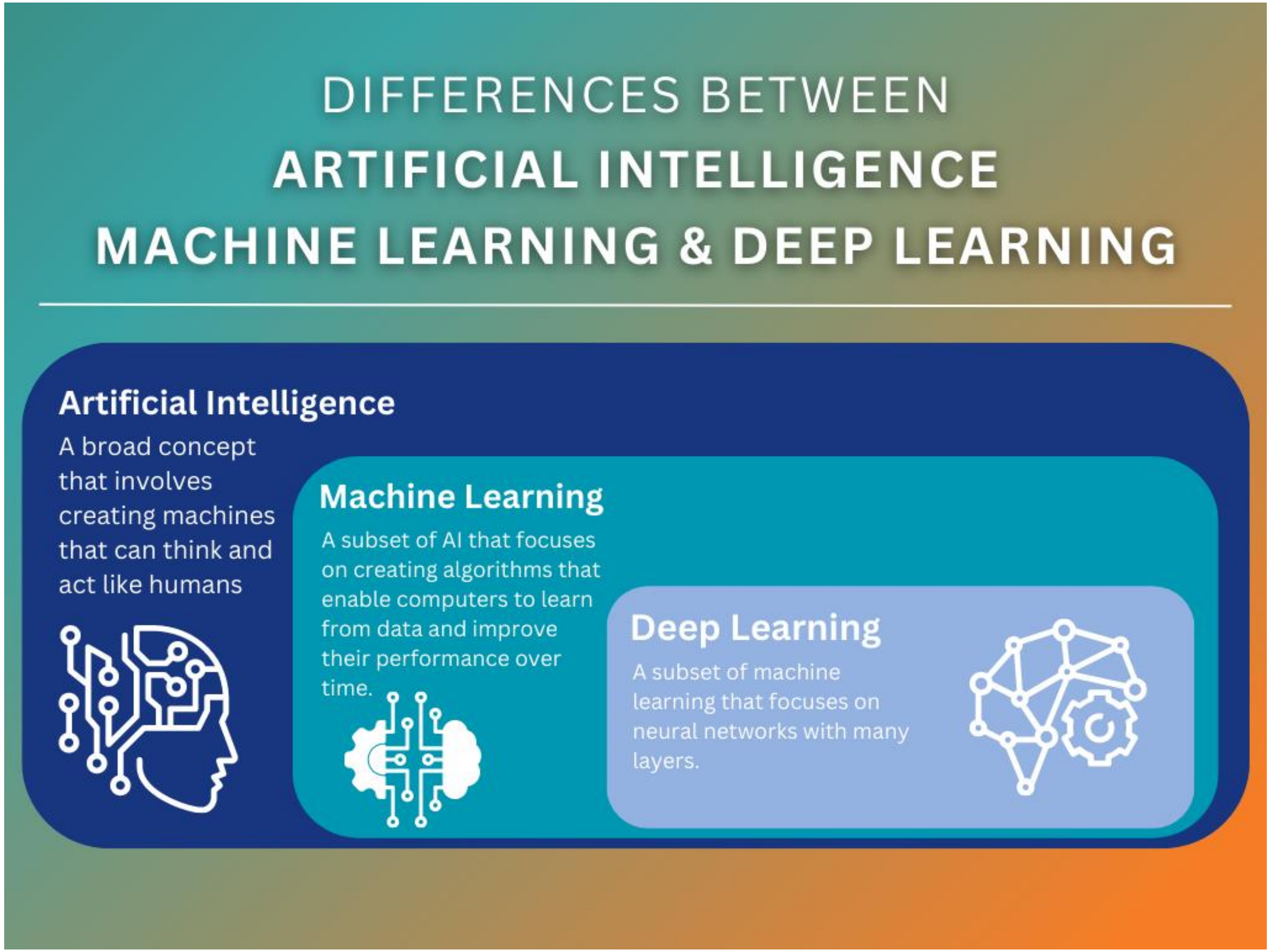
Prerequisites:

To get started with Machine Learning you must be familiar with **Statistics, Linear Algebra, Calculus, Probability, Python and Programming Languages**

Topics covered in this Lecture

- What are AI, ML and DL?
- What is Machine Learning?
- History of Machine Learning

INTRODUCING AI , ML AND DL AND HOW ARE THEY CONNECTED



INTRODUCING AI , ML AND DL AND HOW ARE THEY CONNECTED

Artificial Intelligence

mechanism to incorporate human intelligence into machines through a set of rules(algorithm).

AI is a combination of two words:

"Artificial": is something made by humans or non-natural things and "Intelligence": is the ability to understand or think accordingly.

"AI is the study of training your machine(computers) to mimic a human brain and its thinking capabilities".

AI focuses on 3 major aspects(skills): learning, reasoning, and self-correction to obtain the maximum efficiency possible.

Artificial Intelligence (AI) is a broad concept that involves creating machines that can think and act like humans.

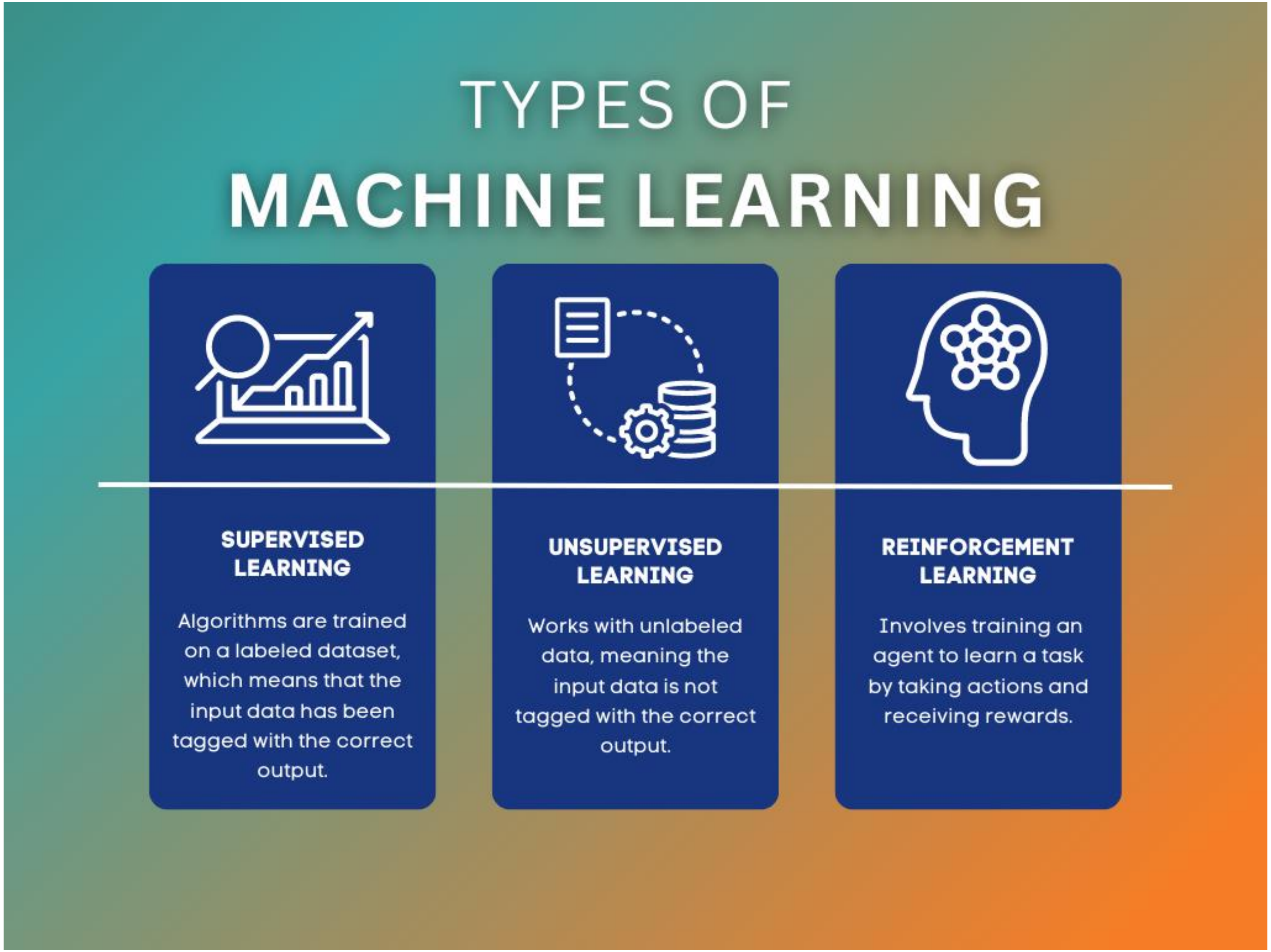
AI systems are designed to perform tasks that usually require human intelligence, such as problem-solving, pattern recognition, learning, and decision-making.

The ultimate goal of AI is to create machines that can perform tasks with minimal human intervention.

INTRODUCING AI , ML AND DL AND HOW ARE THEY CONNECTED

Machine Learning (ML)

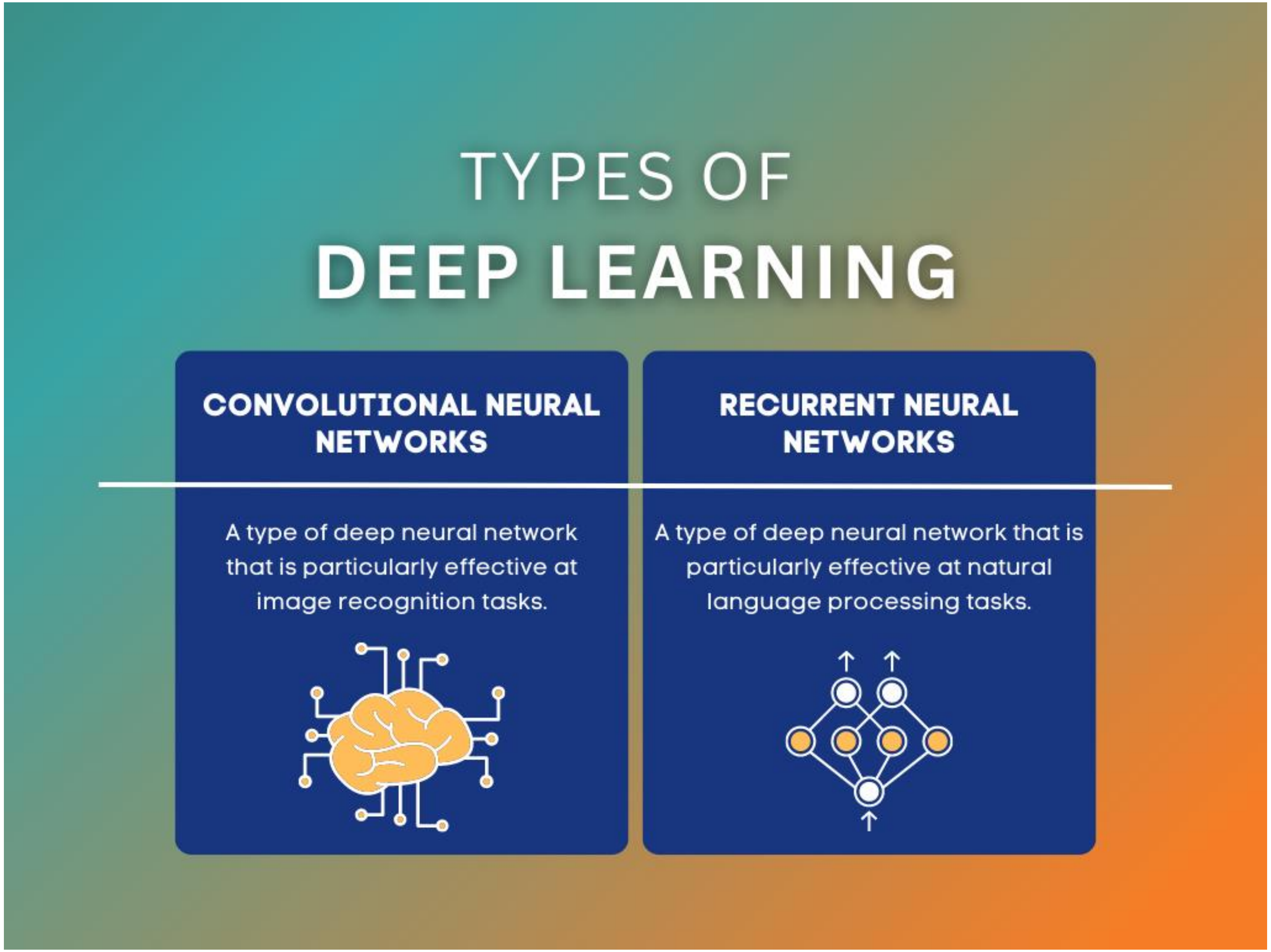
- is a subset of AI
- focuses on creating algorithms that enable computers to learn from data and improve their performance over time.
- ML allows computers to learn and adapt without being explicitly programmed to do so.
- This is accomplished by feeding the algorithms large amounts of data and allowing them to adjust their processes based on the patterns and relationships they discover in the data.


TYPES OF
MACHINE LEARNING
SUPERVISED LEARNING
Algorithms are trained on a labeled dataset, which means that the input data has been tagged with the correct output.
UNSUPERVISED LEARNING
Works with unlabeled data, meaning the input data is not tagged with the correct output.
REINFORCEMENT LEARNING
Involves training an agent to learn a task by taking actions and receiving rewards.

INTRODUCING AI , ML AND DL AND HOW ARE THEY CONNECTED

Deep learning (DL)

- is a subset of machine learning that focuses on neural networks with many layers.
- These deep neural networks are designed to mimic the structure and function of the human brain, allowing computers to process and analyze large amounts of complex, unstructured data.
- DL algorithms focus on **information processing patterns** mechanism to possibly identify the patterns just like our human brain does and classify the information accordingly.
- DL works on larger sets of data when compared to ML and the **prediction mechanism is self-administered by machines**.
- Deep learning algorithms are particularly effective at tasks such as image and speech recognition, natural language processing, and game playing.


TYPES OF
DEEP LEARNING
CONVOLUTIONAL NEURAL NETWORKS
A type of deep neural network that is particularly effective at image recognition tasks.
RECURRENT NEURAL NETWORKS
A type of deep neural network that is particularly effective at natural language processing tasks.

COMPARISON BETWEEN AI, ML, DL

Artificial Intelligence	Machine Learning	Deep Learning
AI stands for Artificial Intelligence and is the study/process that enables machines to mimic human Behaviour through a particular algorithm.	ML stands for Machine Learning, and is the study that uses statistical methods enabling machines to improve with experience.	DL stands for Deep Learning, and is the study that makes use of Neural Networks(similar to neurons present in human brain) to imitate functionality just like a human brain.
AI is the broader family consisting of ML and DL as it's components.	ML is the subset of AI.	DL is the subset of ML.
AI is a computer algorithm that exhibits intelligence through decision-making.	ML is an AI algorithm which allows system to learn from data.	DL is a ML algorithm that uses deep(more than one layer) neural networks to analyze data and provide output accordingly.
Search Trees and much complex math is involved in AI.	If you have a clear idea about the logic(math) involved in behind and you can visualize the complex functionalities like K-Mean, Support Vector Machines, etc., then it defines the ML aspect.	If you are clear about the math involved in it but don't have idea about the features, so you break the complex functionalities into linear/lower dimension features by adding more layers, then it defines the DL aspect.

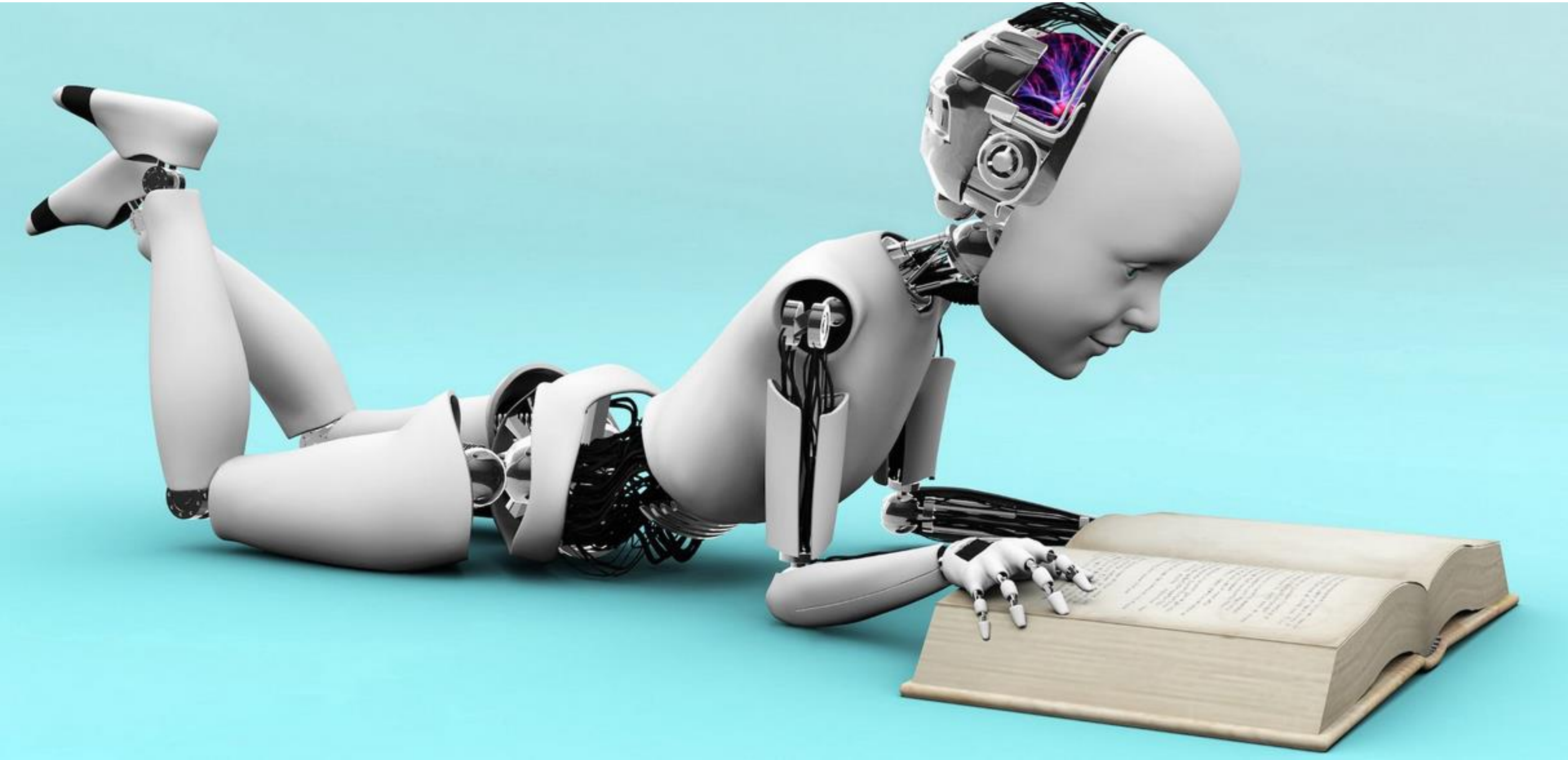
COMPARISON BETWEEN AI, ML, DL

Artificial Intelligence	Machine Learning	Deep Learning
The aim is to basically increase chances of success and not accuracy.	The aim is to increase accuracy not caring much about the success ratio.	It attains the highest rank in terms of accuracy when it is trained with large amount of data.
Three broad categories/types Of AI are: Artificial Narrow Intelligence (ANI), Artificial General Intelligence (AGI) and Artificial Super Intelligence (ASI)	Three broad categories/types Of ML are: Supervised Learning, Unsupervised Learning and Reinforcement Learning	DL can be considered as neural networks with a large number of parameter layers lying in one of the four fundamental network architectures: • Unsupervised Pre-trained Networks, Convolutional Neural Networks, • Recurrent Neural Networks and • Recursive Neural Networks
The efficiency Of AI is basically the efficiency provided by ML and DL respectively.	Less efficient than DL as it can't work for longer dimensions or higher amount of data.	More powerful than ML as it can easily work for larger sets of data.
Examples of AI applications include: Google's AI-Powered Predictions, Ridesharing Apps Like Uber and Lyft, Commercial Flights Use an AI Autopilot, etc.	Examples of ML applications include: Virtual Personal Assistants: Siri, Alexa, Google, etc., Email Spam and Malware Filtering.	Examples of DL applications include: Sentiment based news aggregation, Image analysis and caption generation, etc.

COMPARISON BETWEEN AI, ML, DL

Artificial Intelligence	Machine Learning	Deep Learning
AI refers to the broad field of computer science that focuses on creating intelligent machines that can perform tasks that would normally require human intelligence, such as reasoning, perception, and decision-making.	ML is a subset of AI that focuses on developing algorithms that can learn from data and improve their performance over time without being explicitly programmed.	DL is a subset of ML that focuses on developing deep neural networks that can automatically learn and extract features from data.
AI can be further broken down into various subfields such as robotics, natural language processing, computer vision, expert systems, and more.	ML algorithms can be categorized as supervised, unsupervised, or reinforcement learning. In supervised learning, the algorithm is trained on labeled data, where the desired output is known. In unsupervised learning, the algorithm is trained on unlabeled data, where the desired output is unknown.	DL algorithms are inspired by the structure and function of the human brain, and they are particularly well-suited to tasks such as image and speech recognition.
AI systems can be rule-based, knowledge-based, or data-driven.	In reinforcement learning, the algorithm learns by trial and error, receiving feedback in the form of rewards or punishments.	DL networks consist of multiple layers of interconnected neurons that process data in a hierarchical manner, allowing them to learn increasingly complex representations of the data

WHAT IS MACHINE LEARNING



WHAT IS MACHINE LEARNING

- **Machine learning** is an application of artificial **intelligence** (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

- **Machine learning** focuses on the development of computer programs that can access data and use it to learn for themselves.

OVERVIEW OF MACHINE LEARNING

- The term *machine learning* was coined in 1959 by Arthur Samuel, an American IBMer and pioneer in the field of computer gaming and artificial intelligence.
- Machine learning is about extracting knowledge from data.
- It is a research field at the intersection of statistics, artificial intelligence, and computer science and is also known as predictive analytics or statistical learning.

Arthur Lee Samuel

MACHINE LEARNING -DEFINITIONS

In 1959, **Arthur Samuel**, a computer scientist who pioneered the study of artificial intelligence, described machine learning as

"the study that gives computers the ability to learn without being explicitly programmed."

Alan Turing's seminal paper (**Turing,** 1950)

Machine Learning is an application of artificial intelligence where a computer/machine learns from the past experiences (input data) and makes future predictions. The performance of such a system should be at least human level.

MACHINE LEARNING -DEFINITIONS

A more technical definition given by **Tom M. Mitchell**'s (1997) :

"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."

```
A handwriting recognition learning problem:

Task T: recognizing and classifying handwritten words within images
Performance measure P: percent of words correctly classified,
accuracy
Training experience E: a data-set of handwritten words with given
classifications
```

EVOLUTION OF MACHINE LEARNING

1943 – The First Mathematical Model of a Biological Neuron

1949 – The Hebb Synapse

1950 – The Turing Test

1952 – Machine Learning and the Game of Checkers

1956 – The Birthplace of Artificial Intelligence

1958 – The Perceptron

1963 – A Game of Tic Tac Toe

1965 – The Multilayer Neural Networks Presented

1967 – The Nearest Neighbor Algorithm

Machine Learning Development

1973 – 20th Century AI Winter

1979 – Neocognitron and The Stanford Cart

1981 – Explanation Based-Learning

1982 – The Hopfield Network

EVOLUTION OF MACHINE LEARNING

1985 – The NETTalk
1986 – Restricted Boltzmann Machine
1989 – Boosting for Machine Learning
1991 – The Vanishing Gradient Problem
1992 – Playing Backgammon
1997 – Deep Blue and the Milestone of LSTM
2002 – The Release of Torch
2006 – Deep Belief Network
2009 – ImageNet
2010 – Microsoft's Kinect
2011 – IBM's Watson and Google Brain

EVOLUTION OF MACHINE LEARNING

2012 – ImageNet Classification

2014 – Facebook's DeepFace and Google's Sibyl

2015 – Platform for Machine Learning Algorithms and Toolkit

2016 – AlphaGo Algorithm and Face2Face

2017 – Waymo

2018 – DeepMind's AlphaFold

2020 – GPT-3 and the Rise of No-Code AI

2021 – TrustML and OpenAI's DALL-E

2022 – ChatGPT's Debut, DeepMind's AlphaTensor, and More T2I Models

2023 – LLMs and Computer Vision Reign the Scene

2024 and Beyond

- Quantum Machine Learning (QML)
- Machine Learning Operationalization Management (MLOps)
- Automated Machine Learning (AutoML)
- Robotic Process Automation (RPA)

EVOLUTION OF MACHINE LEARNING

1943 – The First Mathematical Model of a Biological Neuron:

Walter Pitts and Warren McCulloch created the first mathematical model of neural networks in 1943. Their scientific paper, “A Logical Calculus of the Ideas Immanent in Nervous Activity,” was used to create algorithms that mimic human thought processes

1949 – The Hebb Synapse

Canadian psychologist Donald O. Hebb published his book “The Organization of Behavior: A Neuropsychological Theory.” Here, Hebb **theorizes on neuron excitement and communication between neurons** that influenced how psychologists view stimulus processing in the mind.

1950 – The Turing Test

The Turing Test was proposed by Alan Turing, an English computer scientist, as a measure of a computer’s intelligence in 1950. It’s a way to measure artificial intelligence. If someone can’t tell if they’re talking to another person or a computer, then the computer is considered intelligent.

1952 – Machine Learning and the Game of Checkers

English mathematician Arthur Samuel created a computer learning program for playing championship-level computer checkers, which was created for play on the IBM 701. He initiated alpha-beta pruning, a design that measures each side’s chances to win.

Arthur Samuel is the first person to create and popularize the term “machine learning.”

1956 – The Birthplace of Artificial Intelligence

- In machine learning history, the **Dartmouth Workshop** in 1956 is widely considered the founding event of artifici intelligence. Computer scientist John McCarthy invited well-known mathematicians, scientists, and researchers to a six t eight-week workshop. They gathered at Dartmouth College to establish and brainstorm the AI and ML research fields.

1958 – The Perceptron

- The psychologist **Frank Rosenblatt** attempted to build "the first machine capable of producing an original idea" ar subsequently designed the Perceptron, the first neural network ever produced.
- He combined Donald Hebb's model of brain cell interaction with Arthur Samuel's machine-learning efforts.

1963 – A Game of Tic Tac Toe

- Computer Scientist **Donald Michel** designed Machine Educable Noughts And Crosses Engine (MENACE), a large pile c matchboxes that contained several beads and used reinforcement learning to play tic-tac-toe.
- MENACE works a little like a neural network. It is randomly optimized initially, but after playing a few games, it adjusts t favor winning strategies in each situation.

1965 – The Multilayer Neural Networks Presented

Alexey (Oleksii) Ivakhnenko and Valentin Lapa are scientists who worked together to develop the first-ever multi-laye perceptron. It's a hierarchical representation of a neural network that uses a polynomial activation function and is traine using the Group Method of Data Handling (GMDH). Ivakhnenko is often considered the father of deep learning.

1967 – The Nearest Neighbor Algorithm

Thomas Cover and Peter Hart published his “Nearest Neighbor Pattern Classification” in 1967. It laid a foundation for recognizing patterns and regression in machine learning.

1973 – 20th Century AI Winter

The Lighthill report by James Lighthill in 1973 presented a very pessimistic forecast for the development of core aspects in AI research, stating, *“In no part of the field have the discoveries made so far produced the major impact that was then promised*.” This led to reduced AI research funding in all but two British universities, marking a period in machine learning history known as the AI winter.

1979 – Neocognitron and The Stanford Cart

Japanese computer scientist Kunihiko Fukushima published his work on Neocognitron, a hierarchical multilayered network used to detect patterns and inspire convolutional neural networks used for analyzing images. It sparked a revolution in what we now call AI.

1981 – Explanation Based-Learning

Machine learning has come a long way since its inception in 1981. That year, Gerald Dejong introduced the concept of Explanation Based Learning (EBL), in which a computer analyses training data and creates a general rule it can follow by discarding unimportant data. For example, if the software is instructed to concentrate on the queen in chess, it will discard all non-immediate-effect pieces. This laid the foundation for modern supervised learning techniques

1982 – The Hopfield Network

In 1982, American scientist John Hopfield created the Hopfield Network, which is nothing but a recurrent neural network. It's a special kind whose response differs from other neural networks.

The Hopfield network is an associative memory that stores and recalls patterns. It serves as a content-addressable memory system and would be instrumental for further RNN models of the modern deep learning era.

1985 – The NETTalk

In the mid-1980s, Terrence Sejnowski and Charles R. Rosenberg developed NETtalk. It was created to construct simplified models that might shed light on human learning.

Following a knowledge-driven approach, it learns to pronounce written English text by being shown text as input and matching phonetic transcriptions for comparison.

1986 – Restricted Boltzmann Machine

Cognitive scientist Paul Smolensky invented the restricted Boltzmann Machine (RBM) in 1986.

RBM is faster than the traditional Boltzmann Machine because it "restricts" connections between nodes. It's an algorithm useful for dimensionality reduction, classification, regression, collaborative filtering, feature learning, and topic modeling.

1989 – Boosting for Machine Learning

The concept of boosting was first presented in a 1990 paper titled “The Strength of Weak Learnability” by Robert Schapire and Yoav Freund. It marked a necessary development for the evolution of machine learning.

As Schapire states, “A set of weak learners can create a single strong learner.” It simply translates to producing numerous weaker models and combines their predictions to convert them into a single powerful model.

1991 – The Vanishing Gradient Problem

Sepp Hochreiter first identified the vanishing gradient problem. It was a challenge in machine learning development, specifically with deep neural networks.

As the number of layers in a network increases, the value of the derivative decreases until it eventually vanishes altogether. This can make the learning process extremely slow and difficult to manage.

1992 – Playing Backgammon

Researcher Gerald Tesauro created a program based on an artificial neural network capable of playing backgammon with abilities that matched top human players. The backgammon-playing software is called TD-Gammon. It could play at a high level after just a few hours of training, and it continued to improve as it played more games.

1997 – Deep Blue and the Milestone of LSTM

In 1997, IBM’s Deep Blue became the first computer chess-playing system to defeat a reigning world chess champion when it beat Garry Kasparov.

2002 – The Release of Torch

In 2002, the open-source machine learning library Torch was released. This library allowed for more flexibility and customizability than other libraries at the time and quickly became popular among researchers.

2006 – Deep Belief Network

This year marks a remarkable time in the history of machine learning because Geoffrey Hinton created fast-learning algorithms to explain new algorithms that help computers distinguish objects and text in images and videos.

Together with Ruslan Salakhutdinov, Osindero, and Teh, they published the paper "A fast learning algorithm for deep belief nets," in which they stacked multiple RBMs together in layers and called them Deep Belief Networks. The training process is much more efficient for large amounts of data.

2009 – ImageNet

Fei-Fei Li, a professor at Stanford, launched ImageNet, a database of 14 million labeled images, in 2009. It would be a benchmark for deep learning researchers participating in ImageNet competitions (ILSVRC) every year.

2010 – Microsoft's Kinect

A remarkable year for machine learning history is the release of Kinect, a motion-sensing input device for the Xbox 360 gaming console. It can track 20 different human features at 30 times per second.

2011 – IBM’s Watson and Google Brain

Watson is a cognitive system developed by IBM powered by artificial intelligence and natural language processing. In 2011, Watson competed on the game show Jeopardy! against two human competitors and won. This made it the first computer system ever to win a quiz show against humans.

Google’s X Lab team developed a machine learning algorithm named Google Brain. The aim was to create a deep neural network that could learn how to autonomously browse YouTube videos and recognize cats in digital images, just like the human brain.

2012 – ImageNet Classification

AlexNet, a GPU-based CNN model created by Alex Krizhevsky, won Imagenet’s image classification contest with an accuracy of 84%. It significantly improved over the 75 percent success rate of prior models. This victory starts a deep learning revolution that will span the globe.

2014 – Facebook’s DeepFace and Google’s Sibyl

Facebook developed DeepFace, a deep learning facial software algorithm that can recognize and verify individuals on photos with human accuracy. It’s one of the advanced computer algorithms that can identify human faces with an accuracy of 97.35%.

2015 – Platform for Machine Learning Algorithms and Toolkit

Amazon launches its machine learning platform. The e-commerce giant makes machine learning accessible to anyone with an Amazon Web Services (AWS) account. The platform provides a set of tools and algorithms for data scientists to build and train models.

Microsoft had also developed the Distributed Machine Learning Toolkit, which allowed for the efficient sharing of machine learning problems across multiple computers.

2016 – AlphaGo Algorithm and Face2Face

Go is an ancient Chinese board game with so many possible moves at each step that future positions are hard to predict. When the AlphaGo algorithm was developed in March 2016, it shocked the world by defeating one of the best Go players, Lee Sedol.

2017 – Waymo

Waymo became the first self-driving car company to operate without human intervention. The company's vehicles have now driven over 5 million miles on public roads, with human drivers only intervening when necessary

2018 – DeepMind's AlphaFold

After creating AlphaGo, the team took the first step in developing algorithms for problems exactly like protein folding. AlphaFold was built to predict the 3D shapes of proteins, the fundamental molecules of life.

2020 – GPT-3 and the Rise of No-Code AI

When the world was grappling with the pandemic in 2020, OpenAI created an artificial intelligence algorithm, GPT-3, that could generate human-like text. In its time, it's the most advanced language model in the world, using 175 billion parameters and Microsoft Azure's AI supercomputer for training.

2021 – TrustML and OpenAI's DALL-E

Indian American computer scientist Himabindu "Hima" Lakkaraju not only co-founded the Trustworthy ML Initiative (TrustML), but she also leads the AI4LIFE research group at Harvard.

Introduced in January 2021, DALL-E is a variant of GPT-3, a language-processing model from OpenAI. It delves into generating images from text, adding a whole new dimension to language processing. Powered by the transformer neural network, DALL-E is reshaping how we interact with AI technology.

2022 – ChatGPT's Debut, DeepMind's AlphaTensor, and More T2I Models

OpenAI unveiled an early ChatGPT demo on November 30, 2022. The chatbot went viral on social media, showcasing its versatility. From travel planning to writing fables and coding, users marveled at its capabilities.

Within five days, it amassed over a million users.

In October, DeepMind introduced AlphaTensor. According to a DeepMind blog, AlphaTensor extends AlphaZero, which excelled in chess and Go. This new work progresses from games to tackling unsolved mathematical problems.

2023 – LLMs and Computer Vision Reign the Scene

In 2023, we saw the rise of LLMs or Large Language Models, with GPT-4 being launched on March 14, 2023.

We also saw the evolution of LLMs into multimodal systems, or what they call Multimodal LLMs. Notable MLLMs include OpenAI's GPT-4 Vision and Google DeepMind's Gemini. These allow users to interact with the system using text, images, and speech.

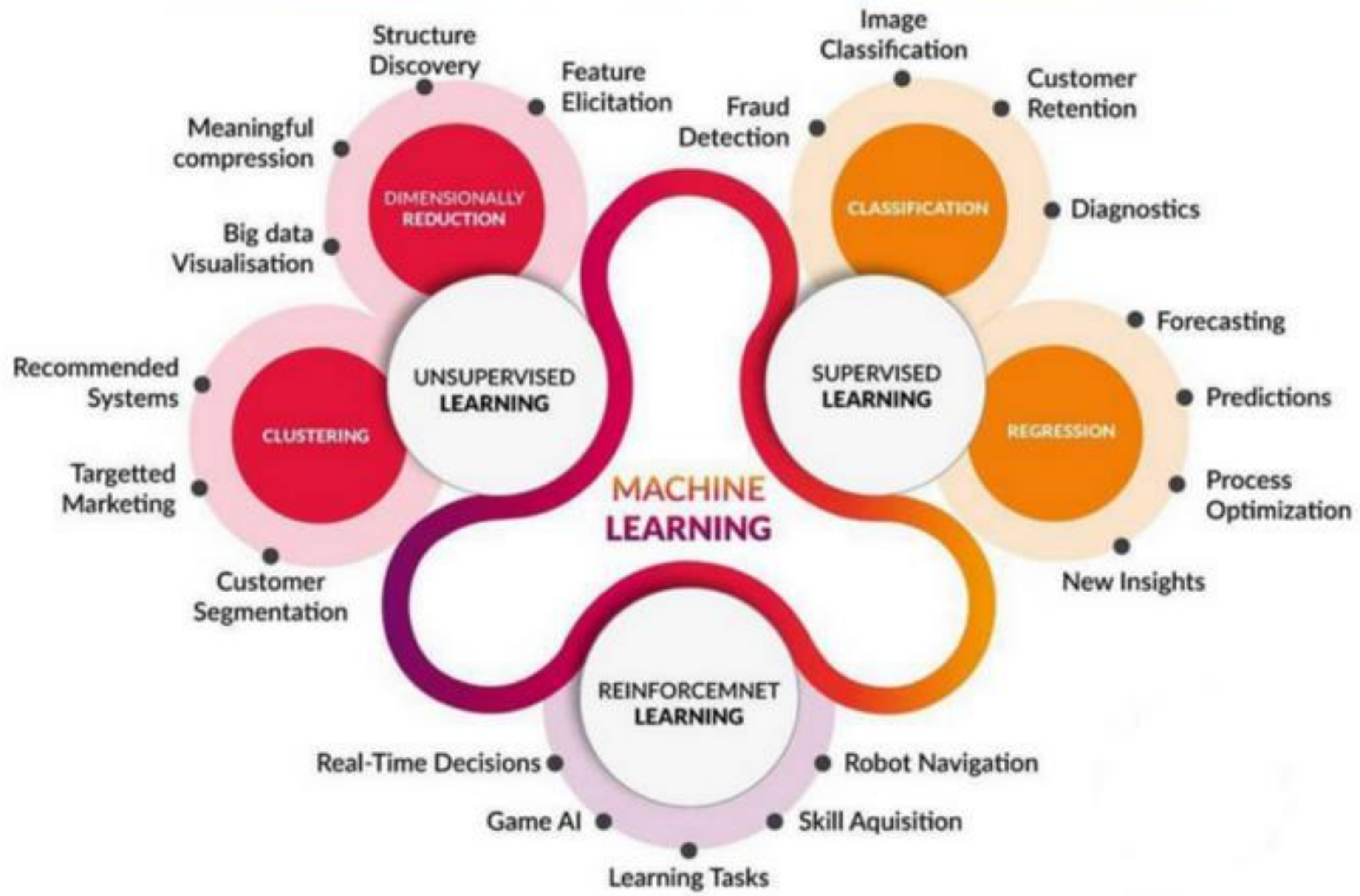
Additionally, computer vision continued to make significant progress. In September 2023, Google introduced Vision Transformer, a deep learning-based model that performs image recognition tasks better than previous methods

2024 and Beyond

In the future, we can expect machine learning improvements on:

- **Quantum Machine Learning (QML)**
- Quantum computers lead to faster processing of data, enhancing the algorithm's ability to analyze and draw meaningful insights from data sets.
- **Machine Learning Operationalization Management (MLOps)**
- This helps machine learning algorithms deployed in production to perform optimally and reliably.
- **Automated Machine Learning (AutoML)**
- AutoML will make the process of training data easier, helping with data labeling and reducing human error in operations.
- **Robotic Process Automation (RPA)**
- A data-driven approach is needed before an RPA bot can process it, and machine learning will help it produce fewer errors.

Machine learning applications



Applications of Machine Learning

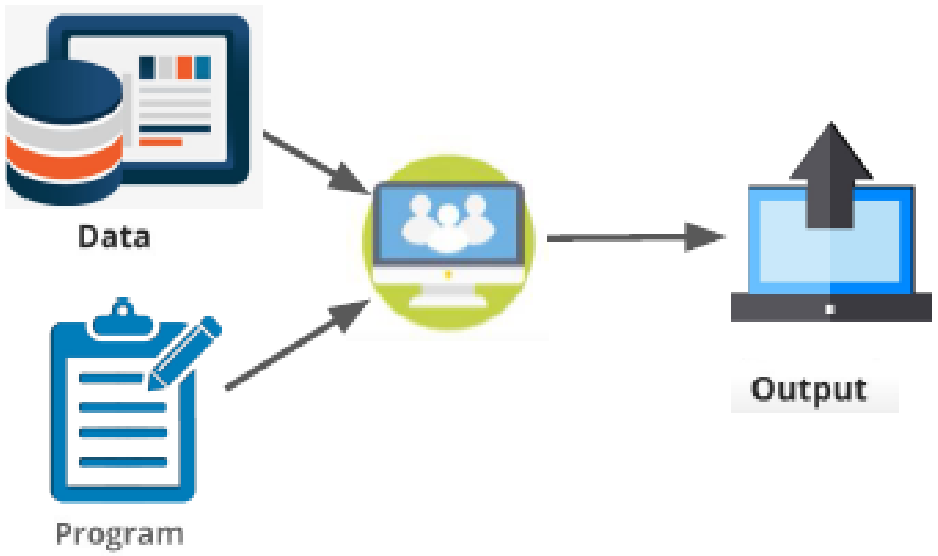
1. **Web search**: ranking page based on what you are most likely to click on.
2. **Computational** biology: rational design drugs in the computer based on past experiments.
3. **Finance**: decide who to send what credit card offers to. Evaluation of risk on credit offers. How to decide where to invest money.
4. **E-commerce**: Predicting customer churn. Whether or not a transaction is fraudulent.
5. **Space exploration**: space probes and radio astronomy.
6. **Robotics**: how to handle uncertainty in new environments. Autonomous. Self-driving car.

7. **Information extraction**: Ask questions over databases across the web.
8. **Social networks**: Data on relationships and preferences. Machine learning to extract value from data.
9. **Debugging**: Use in computer science problems like debugging. Labor intensive process. Could suggest where the bug could be.

Traditional Vs Machine learning

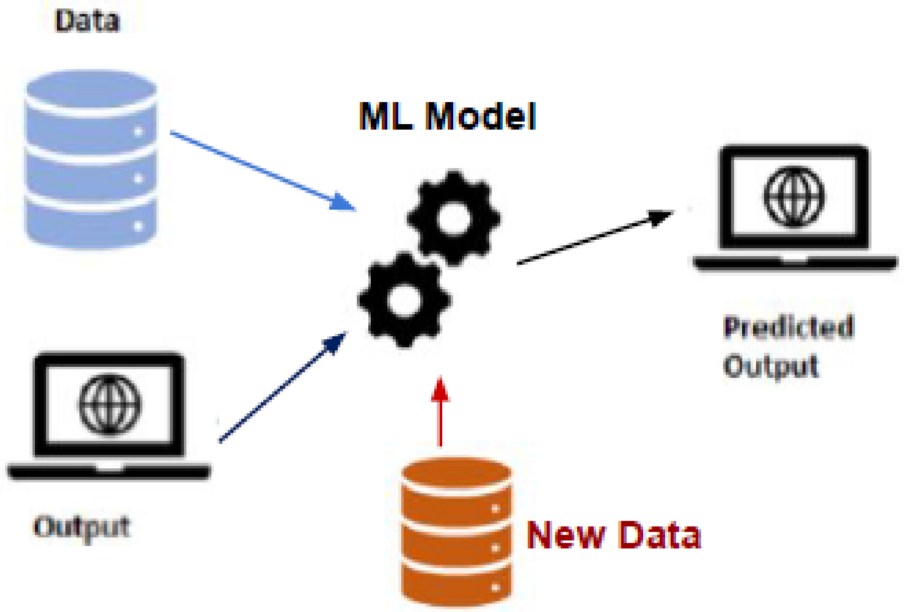
Traditional Programming

Traditional Programming: You code the behavior of the program



Machine Learning

Machine Learning: You leave a lot of that to the machine to learn from data



Data set sample

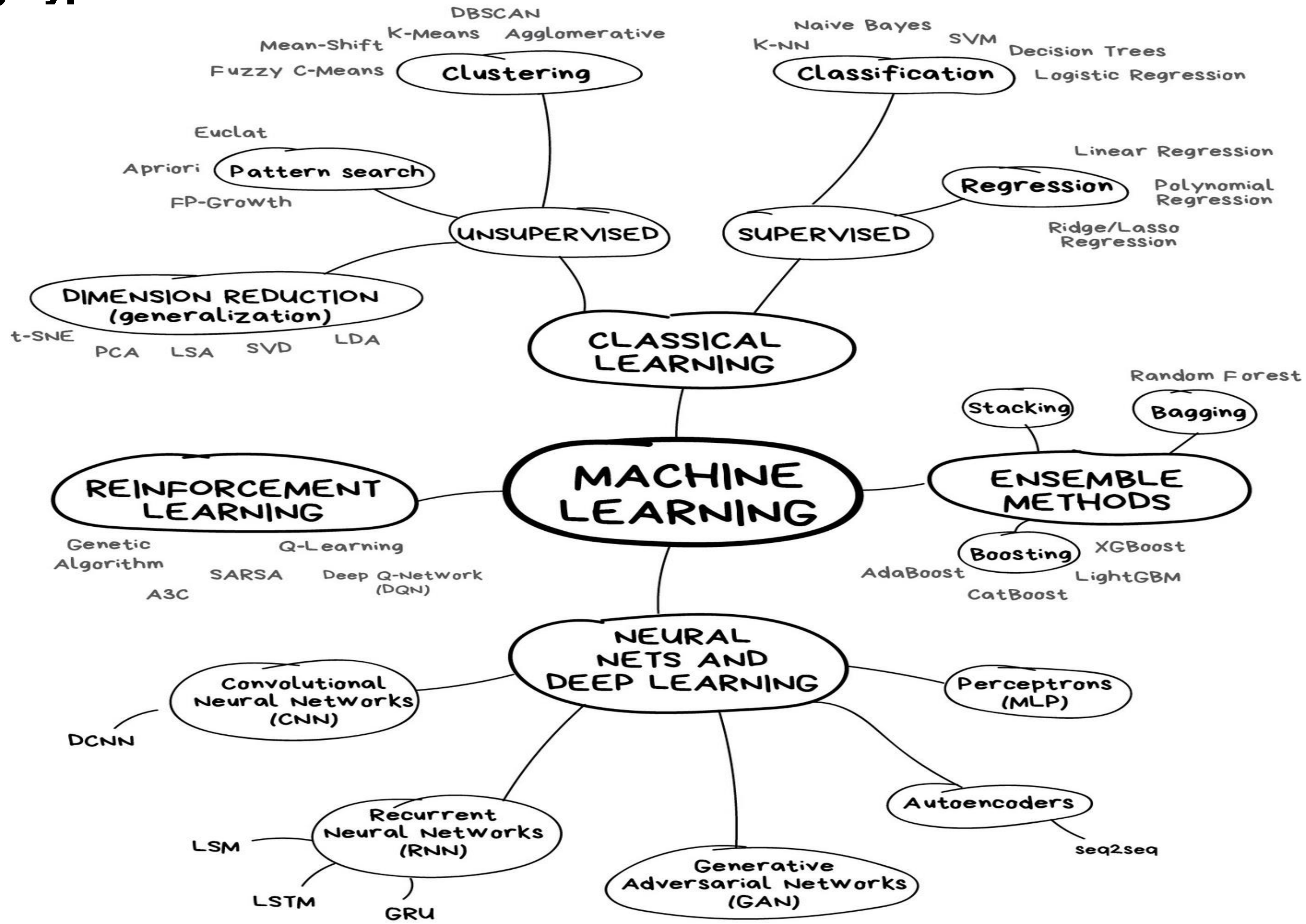
Example Data About Customers							
	Age	Number of cars Owned	Own House	Number of Children	Marital Status	Owns a Dog	Bought a Boat
1	66	1	yes	2	widowed	no	yes
2	52	2	yes	3	married	no	yes
3	22	0	no	0	married	yes	no
4	25	1	no	1	single	no	no
5	44	0	no	2	divorced	yes	no
6	39	1	yes	2	married	yes	no
7	26	1	no	2	single	no	no
8	40	3	yes	1	married	yes	no
9	53	2	yes	2	divorced	no	yes
10	64	2	yes	3	divorced	no	no
11	58	2	yes	2	married	yes	yes
12	33	1	no	1	single	no	no

Data Set about Customers

Attributes/Features

Records/Tuples of Customers who have bought Boats

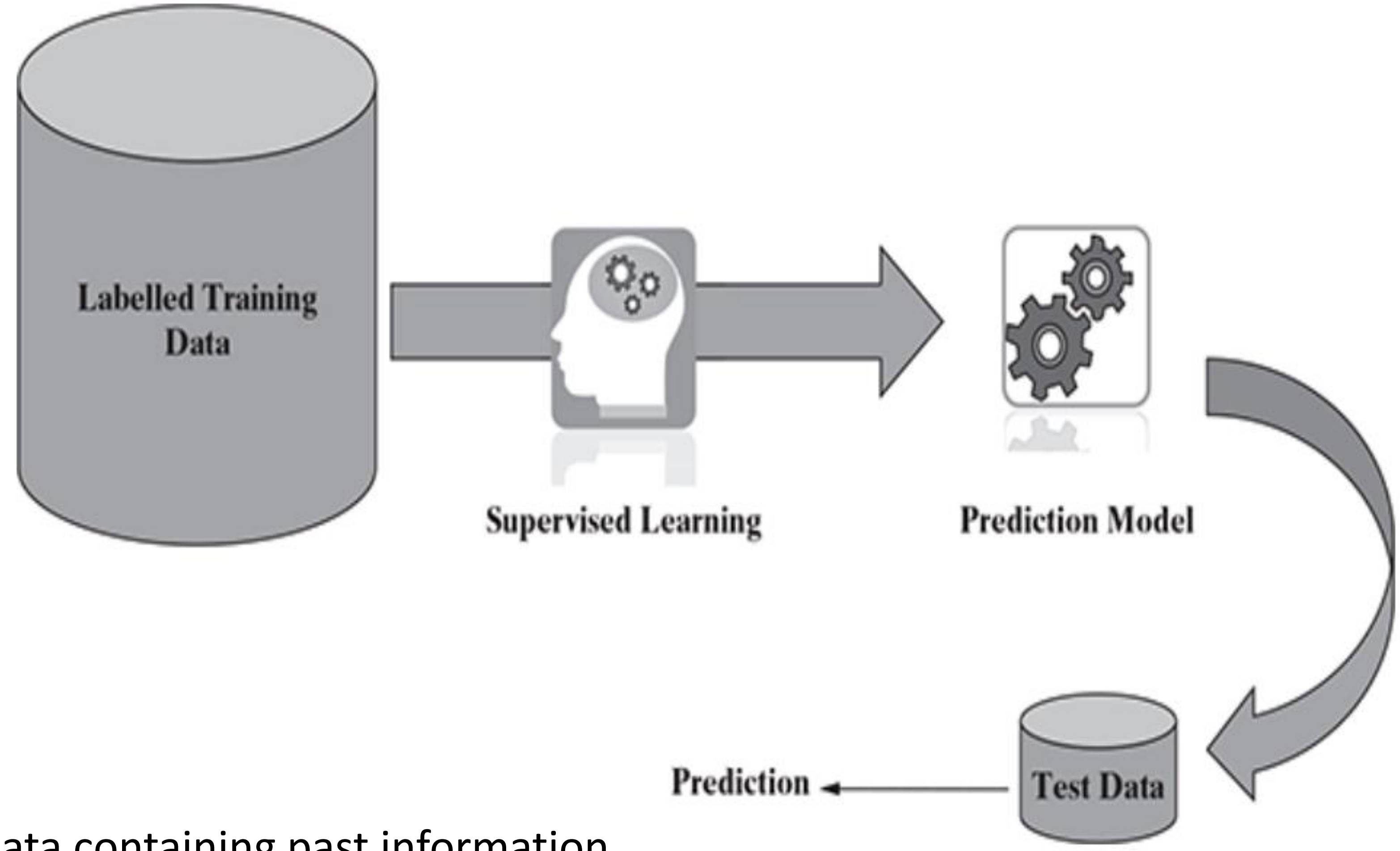
Machine Learning Types



SUPERVISED LEARNING

- Learning from examples/ Labelled data
- Training set is given which acts as an example/ experience for the classes.
- System finds a description for each class (Classification Rule)
- Once the description has been formulated, it is used to predict the class for the new object.

SUPERVISED LEARNING



- Labeled training data containing past information comes as an input.
- Based on the training data, the machine builds a predictive model that can be used on test data to assign a label for each record in the test data.

- It is a machine learning task of learning a function that maps an input to an output based on example input-output pairs.
- It infers a function from *labeled training data* consisting of a set of *training examples*.
- In supervised learning, each example is a *pair* consisting of an input object (typically a vector) and a desired output value (also called the *supervisory signal*).
- A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.

Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

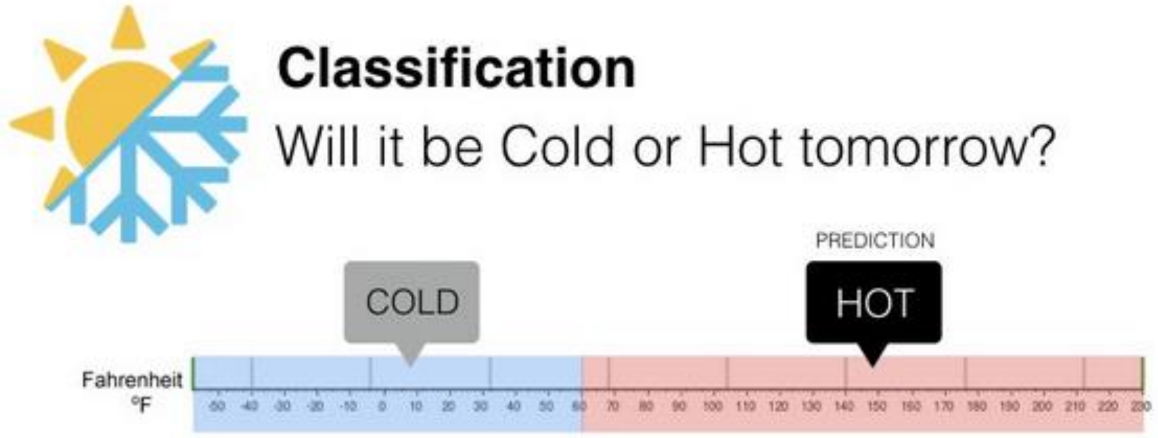
$$Y = f(X)$$

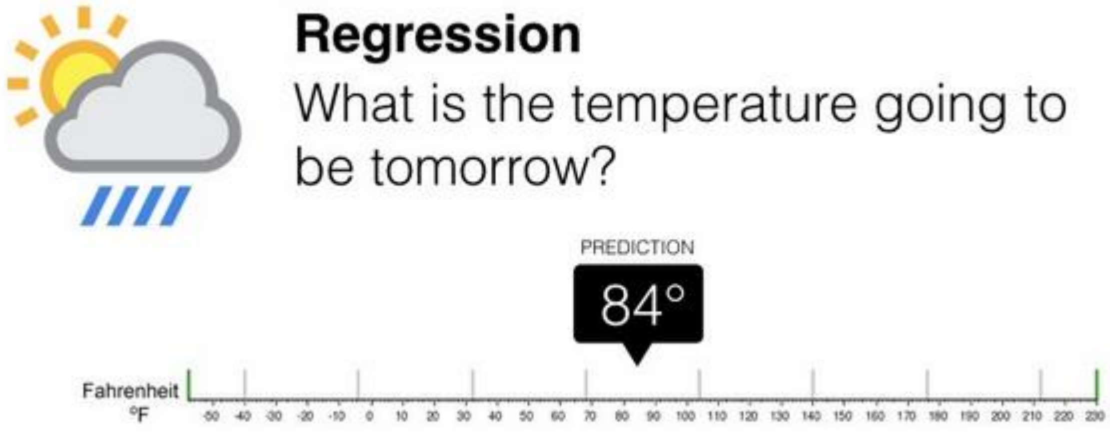
The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

- Supervised learning is the most popular paradigm for performing machine learning operations.
- It is widely used for datasets where there is a precise mapping between input-output data.
- The dataset, in this case, is **labeled**, meaning that the algorithm identifies the features explicitly and carries out predictions or classification accordingly.
- As the training period progresses, the algorithm can identify the relationships between the two variables such that we can predict a new outcome.

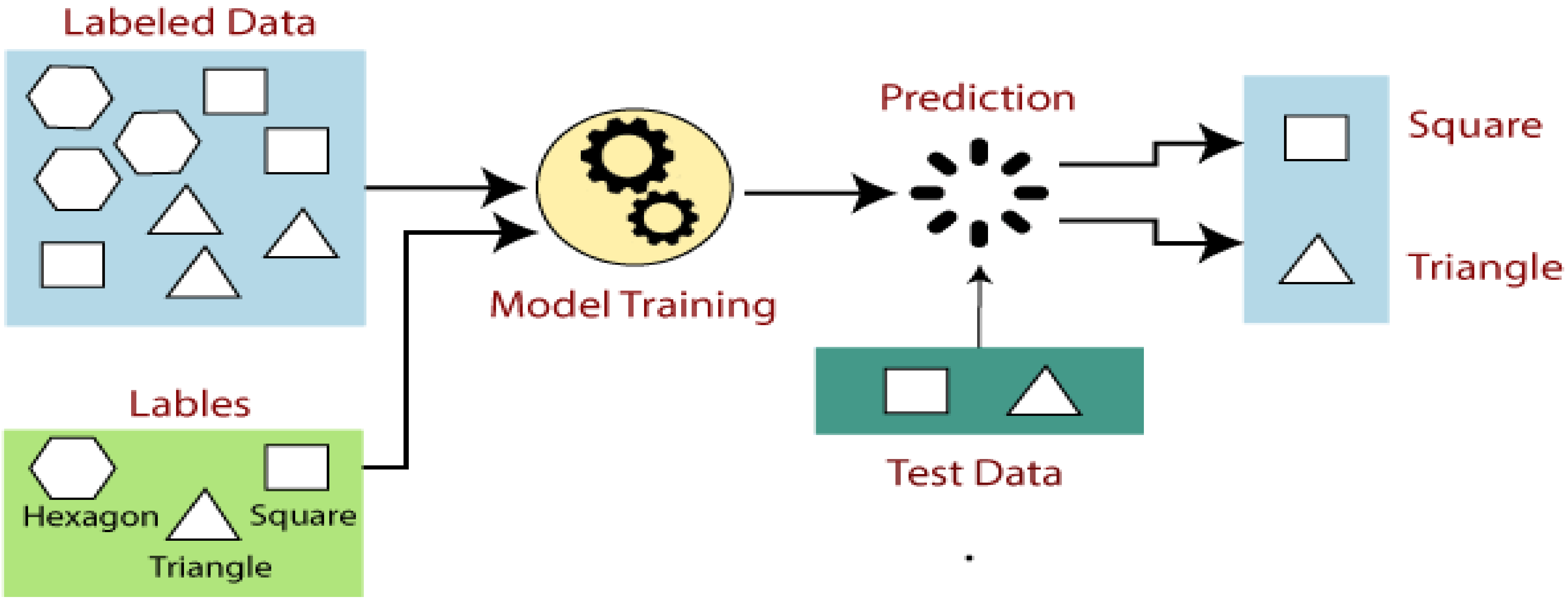
SUPERVISED LEARNING METHODS

1. Classification
2. Regression





CLASSIFICATION- SUPERVISED LEARNING



CLASSIFICATION

- **Classification**
  - predicts categorical class labels (discrete or nominal)
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data

Classification is the process of finding a **model** (or function) that describes and distinguishes data classes or concepts.

The model are derived based on the analysis of a set of

- **training data** (i.e., data objects for which the class labels are known).
- The model is used to predict the class label of objects for which the class label is unknown.

***"How is the derived model presented?" ***

The derived model may be represented in various forms, such as

- *classification rules* (i.e., *IF-THEN rules*),
- *decision trees*
- *Mathematical Formulae*, or
- *neural networks*

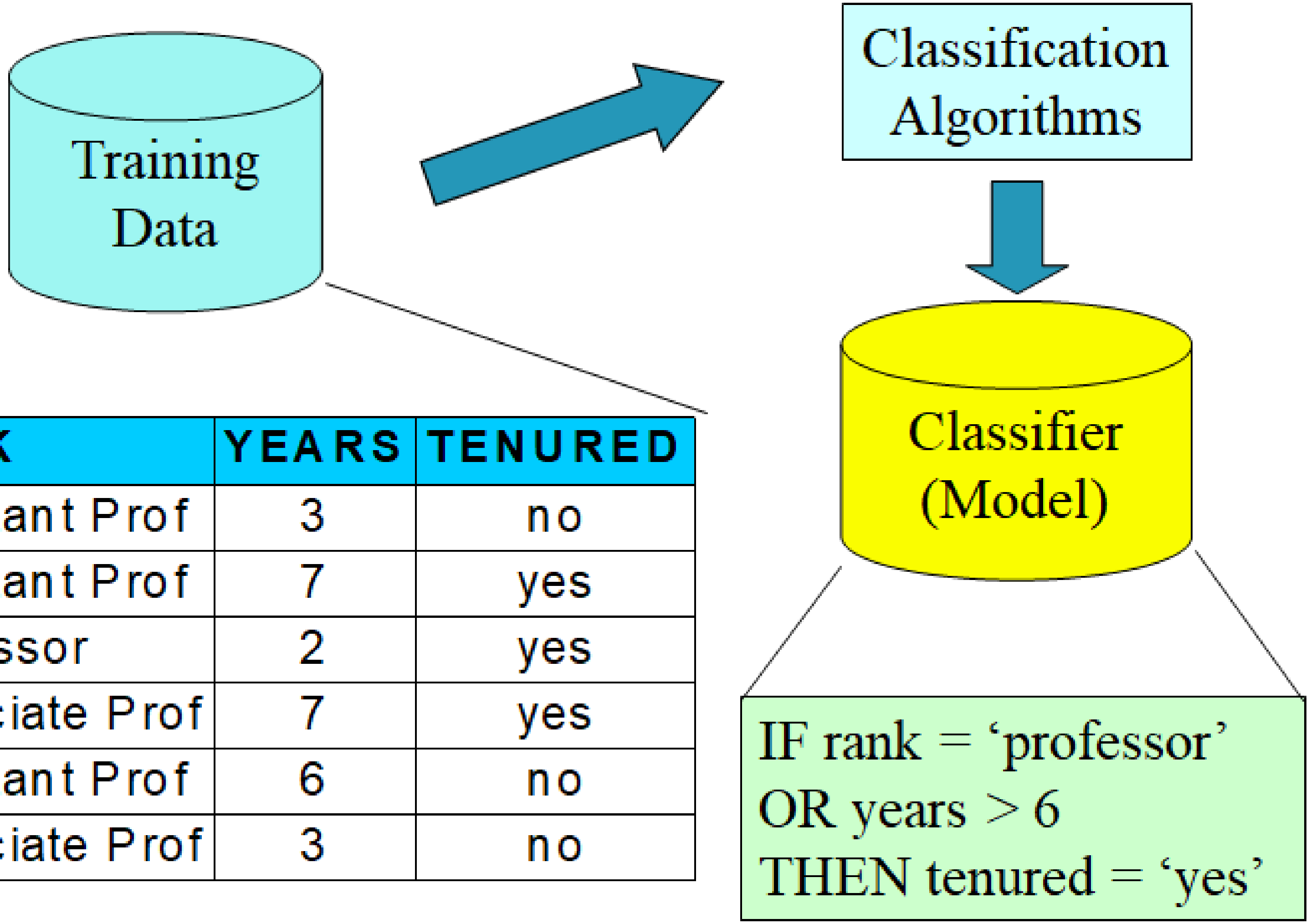
Other methods for constructing classification models, such as

- Naïve Bayesian classification
- support vector machines, and
- *k*-nearest-neighbor classification.

CLASSIFICATION—A TWO-STEP PROCESS

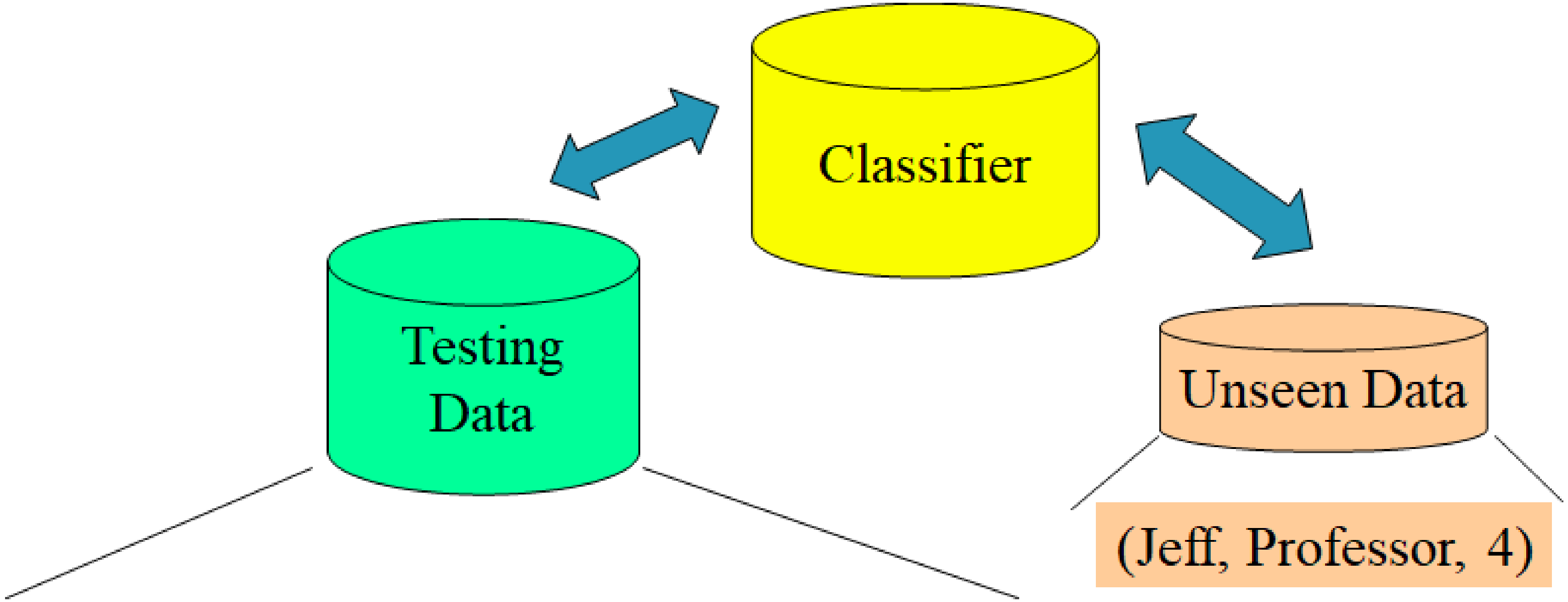
- **Model construction:** describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the **class label attribute**
  - The set of tuples used for model construction is a **training set**
  - The model is represented as classification rules, decision trees, or mathematical formulae
- **Model usage:** for classifying future or unknown objects
  - **Estimate the accuracy** of the model
    - The known label of the test sample is compared with the classified result from the model
    - **Accuracy** rate is the percentage of test set samples that are correctly classified by the model
    - **Test set** is independent of the training set (otherwise overfitting)
  - If the accuracy is acceptable, use the model to **classify new data**
- Note: If *the test set* is used to select models, it is called a **validation (test) set**

PROCESS (1): MODEL CONSTRUCTION

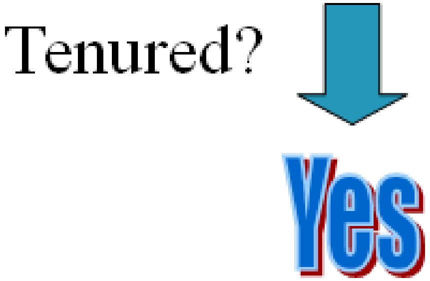


NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

PROCESS (2): USING THE MODEL IN PREDICTION



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes



CLASSIFICATION ALGORITHMS CATEGORIES:

1) Statistical based algorithms: Based directly on the use of statistical information

Ex: Regression (Division and Prediction), Bayesian classification

2) Distance based algorithms: Use similarity or distance measure to perform classification

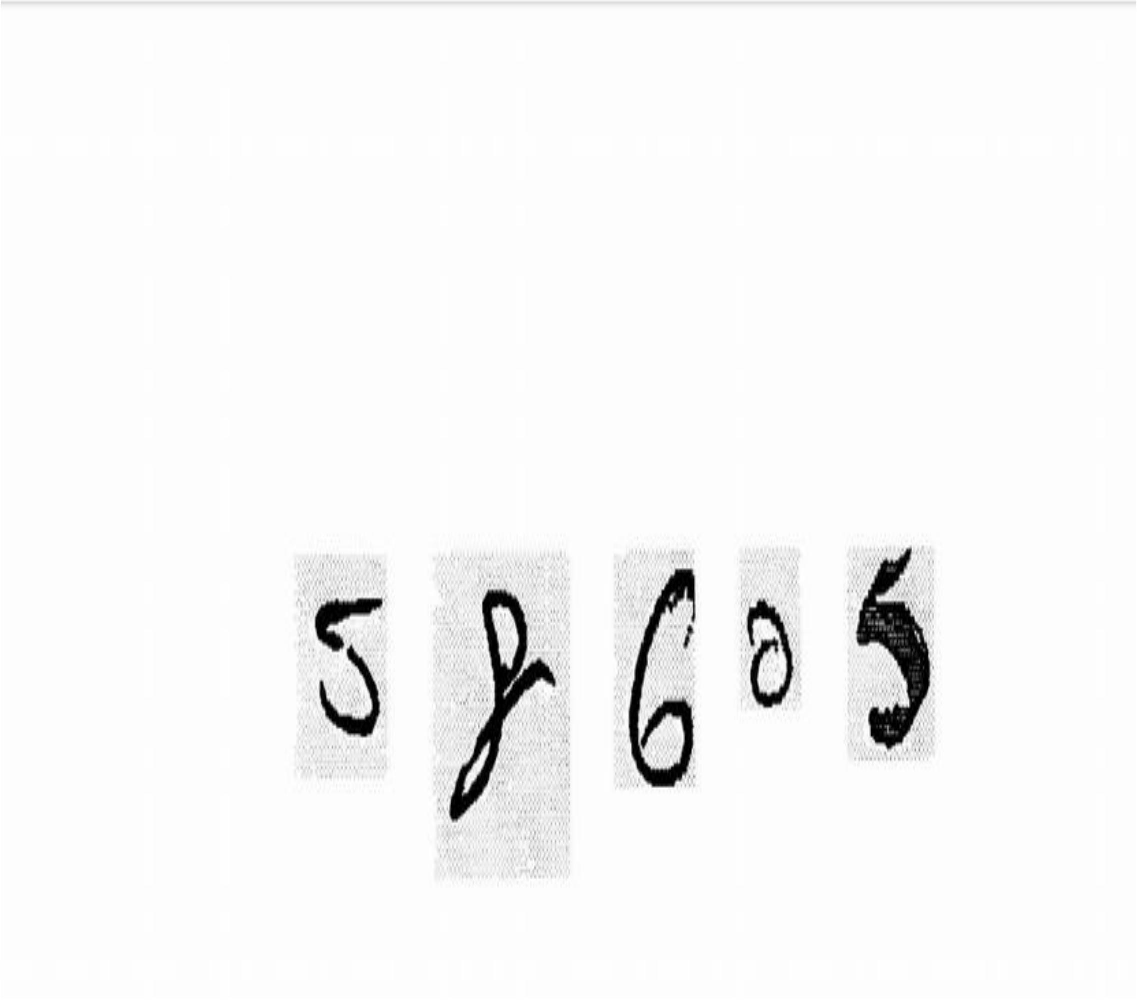
Ex: Simple approach, KNN(K Nearest Neighbor)

3) Decision tree based algorithms: Uses Decision tree structure perform classification

Ex: ID3 ,C4.5, CART, Scalable DT Techniques SPRINT

EXAMPLE 1: CLASSIFICATION

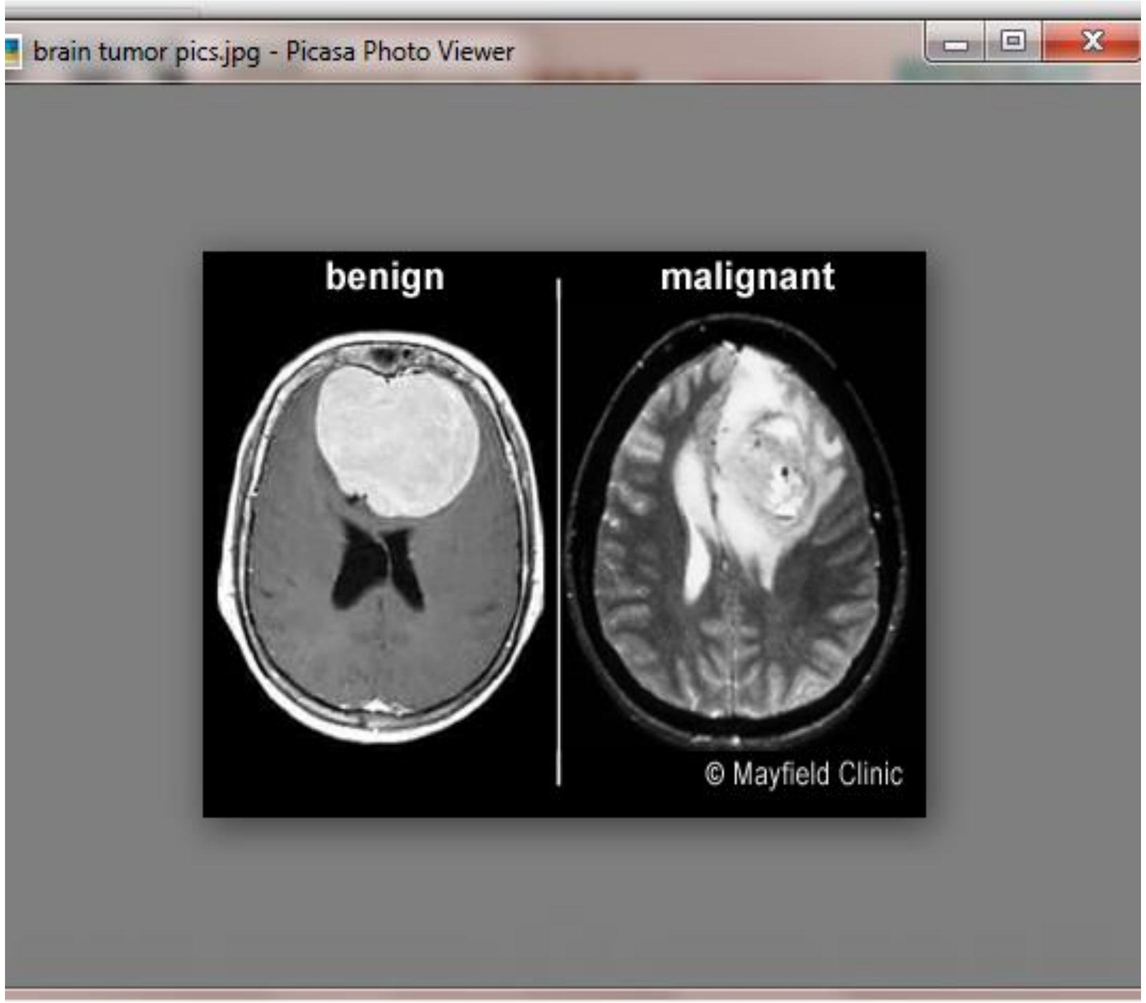
- Identifying the numbers from handwritten digits



EXAMPLE 2: CLASSIFICATION

- Determining whether the Tumor is Benign or Malignant

- Benign tumor-noncancerous
- Malignant- cancerous



EXAMPLE 3: CLASSIFICATION

- **Detecting Fraudulent activity in Credit Card Transactions**

EXAMPLE 4:CLASSIFICATION

- Identifying whether a mail is a spam or a ham.

EXAMPLE 5: CLASSIFICATION

- Classification of Iris Flowers into several species using Physical measurements of the Flower

2) REGRESSION

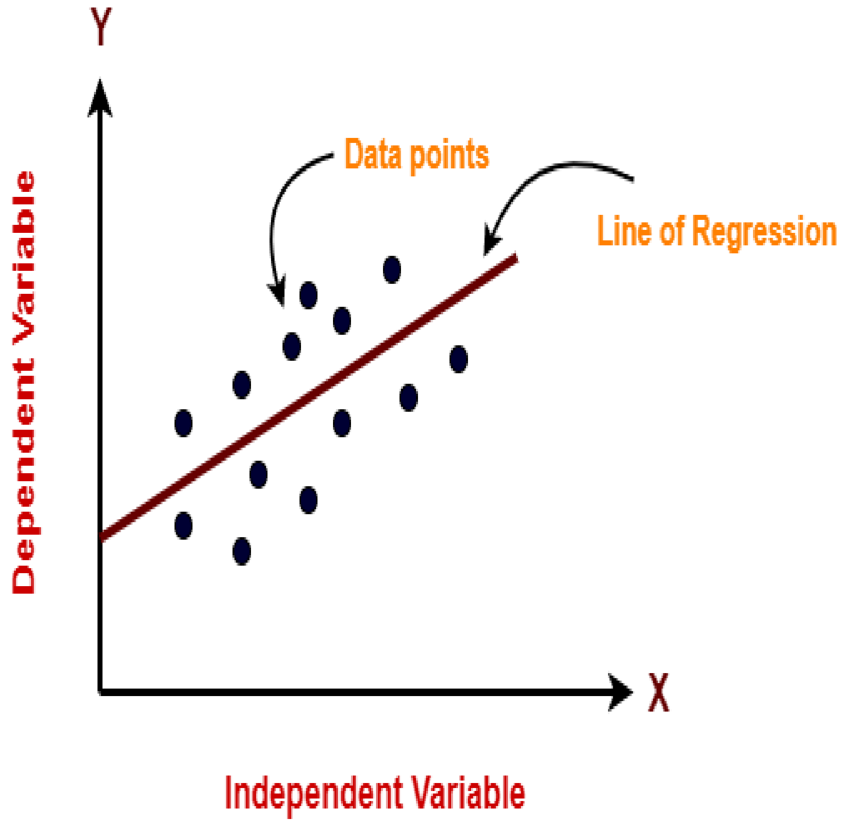
- It allows one to make predictions from data by **learning** the relationship between features of your data and some observed, continuous-valued response.
- For Ex., Predicting prices of a house given the features of house like size, price etc is one of the common **examples** of **Regression**

- **Regression algorithm**
- Simple Linear Regression(Statistical Based Algorithm)
- Multiple Linear Regression

regression algorithm

- Linear Regression(Statistical Based Algorithm)

Regression analysis consists of a set of machine learning methods that **allow us to predict a continuous outcome variable** (y) based on the value of one or multiple predictor variables (x). ... It assumes a linear relationship between the outcome and the predictor variables



EXAMPLE :

- Predicting the price of a house given the details such as the location, apartment / row house / independent house, built up area, corner house

EXAMPLE 7:

- Predicting the yield of a corn in the Farm given the attributes like previous yields, weather, number of employees working on the Farm.

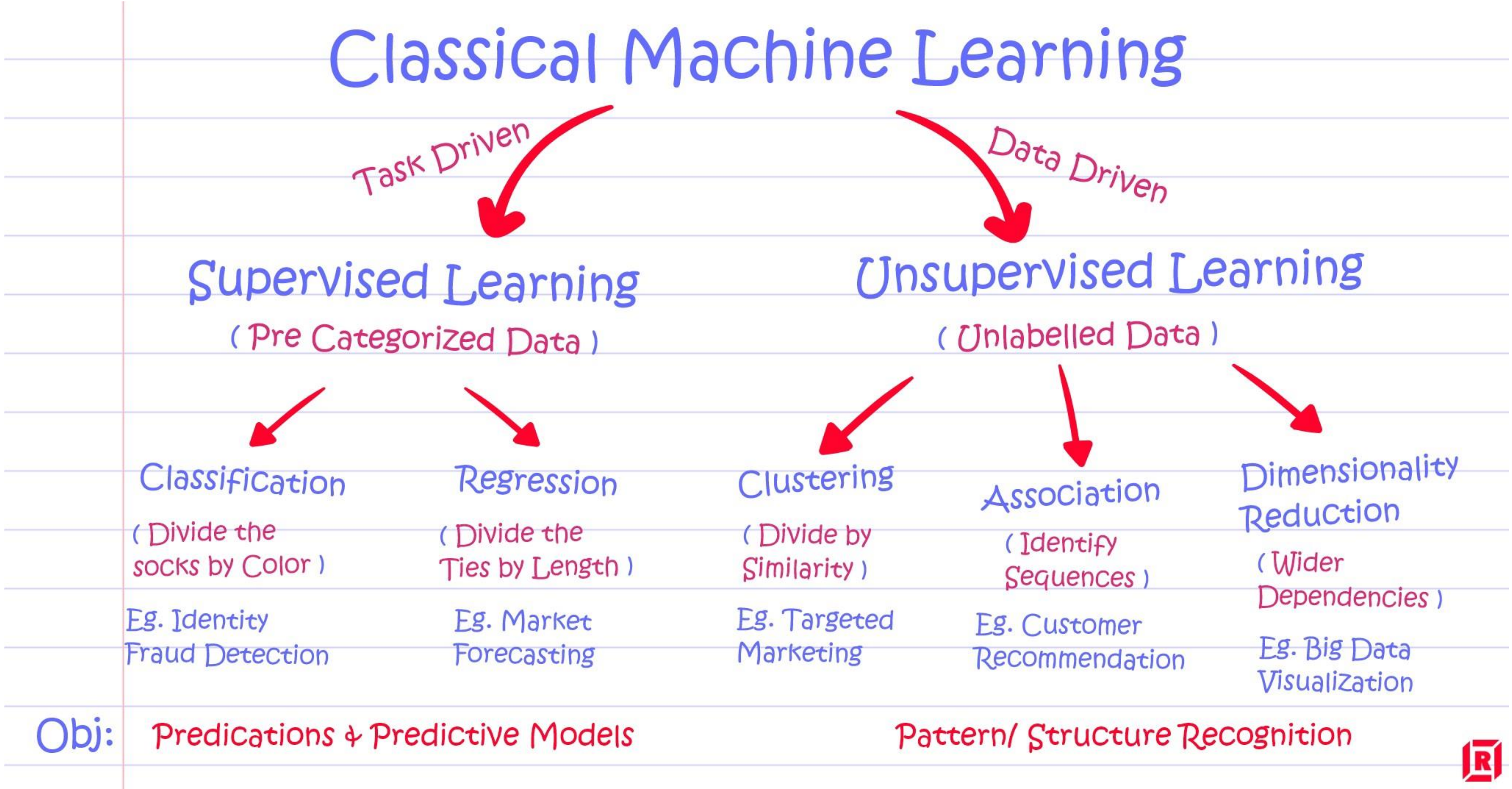
Find out?

- Game match fixing for win or loss
- Cancer cell prediction
- Weather forecasting
- Rainfall forecasting
- Market trend prediction
- Tissue is good or bad
- Predicting the age of a person
- Predicting the nationality of a person
- Predicting the gender of a person using his/her handwriting.
- Predict the number of copies of music album that will be sold in the next month.

2) UNSUPERVISED LEARNING (UL)

- learning from observation and discovery
- is a type of algorithm that learns patterns from untagged/ Unlabelled data
- The machine is forced to build a compact internal representation of its world. In contrast to supervised learning (SL) where data is tagged by a human, e.g. as "car" or "fish" etc,
- UL exhibits self-organization that captures patterns as neuronal predilection or probability densities.
- In Unsupervised Learning, only the input data is known, and no known output data is given to the algorithm.
- For Ex., Identifying topics in a set of blog posts, Segmenting Customers into groups with similar preferences

TYPES OF UNSUPERVISED LEARNING



UNSUPERVISED LEARNING TYPES

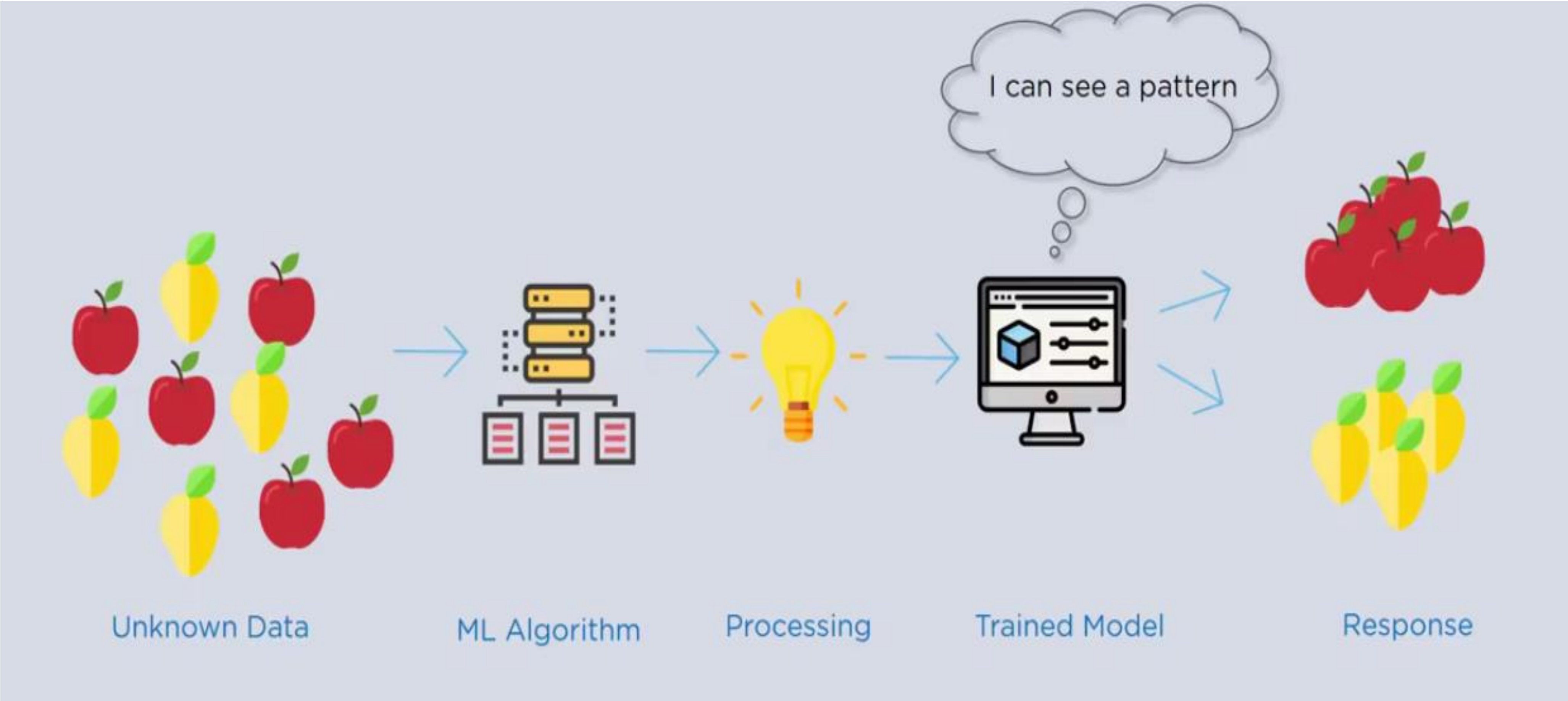
sample Cluster/group

1) Clustering

- **Cluster:** A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- **Cluster analysis (or *clustering, data segmentation, …*)**
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- **Unsupervised learning**: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- **Typical applications**
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

UNSUPERVISED LEARNING

- Learning from observation and discovery



MAJOR CLUSTERING APPROACHES

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

MAJOR CLUSTERING APPROACHES

- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus

2) ASSOCIATION:

- An unsupervised learning method, used to find **the relationships between variables in the large database.**
- It determines the set of items that occur together in the dataset.
- Association rule makes marketing strategy more effective.
- Such as People who buy X items (suppose bread) also tend to purchase Y (Butter/Jam) items. A typical example of an Association rule is Market Basket Analysis.

APPLICATIONS OF ASSOCIATION RULE LEARNING

Market Basket Analysis:

used by big retailers to determine the association between items.

Medical Diagnosis:

With the help of association rules, patients can be cured easily, as it helps in identifying the probability of illness for a particular disease.

Protein Sequence:

The association rules help in determining the synthesis of artificial Proteins.

3) DIMENSIONALITY REDUCTION

- Refers to techniques for reducing the number of input variables in training data.
- When dealing with high dimensional data, it is often useful to reduce the dimensionality by projecting the data to a lower dimensional subspace which captures the “essence” of the data. This is called dimensionality reduction.
- Dimensionality reduction is a data preparation technique performed on data before modeling.
- It might be performed after data cleaning and data scaling and before training a predictive model.
- any dimensionality reduction performed on training data must also be performed on new data, such as a test dataset, validation dataset, and data when predicting with the final model.

Dimensionality Reduction Techniques: Examples

- Manifold Learning (t-SNE, UMAP)
- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Sequential Non-negative Matrix Factorization (NMF)
- Linear Discriminant Analysis (LDA)
- Generalized Discriminant Analysis (GDA)
- Missing Values Ratio (MVR): Threshold Setting
- Low Variance Filter
- High Correlation Filter
- Forward Feature Construction
- Backward Feature Elimination
- Autoencoders

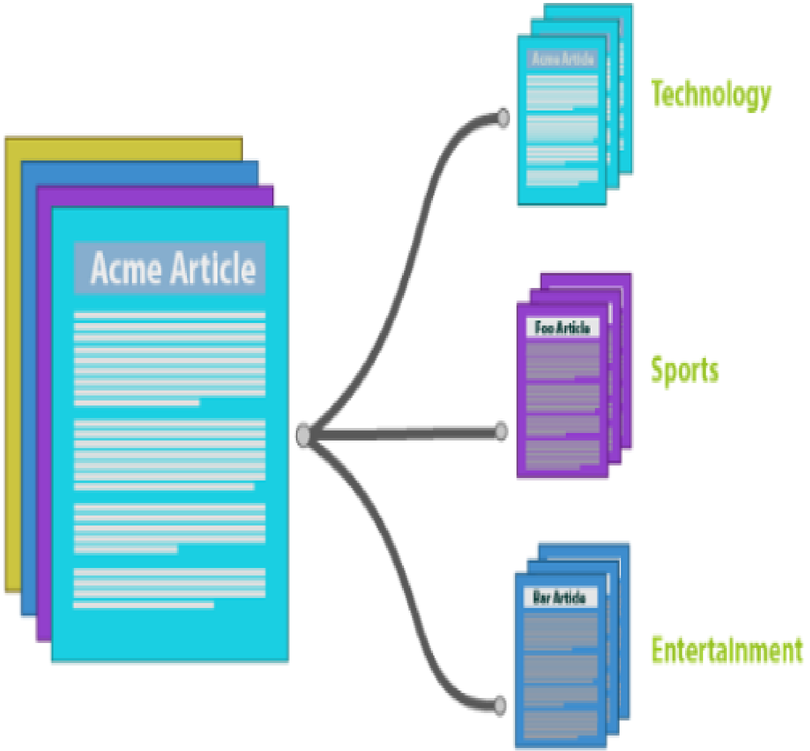
ADVANTAGES OF DIMENSIONALITY REDUCTION

- helps in data compression, and hence reduced storage space.
- It reduces computation time.
- It also helps remove redundant features, if any.
- It takes care of multicollinearity that improves model performance. It removes redundant features. For example, there is no point in storing a value in two different units (meters and inches).
- Reducing the dimensions of data to 2D or 3D may allow us to plot and visualize it precisely. You can then observe patterns more clearly.
- It is helpful in noise removal also and as a result of that, we can improve the performance of models.

UNSUPERVISED LEARNING EXAMPLES:

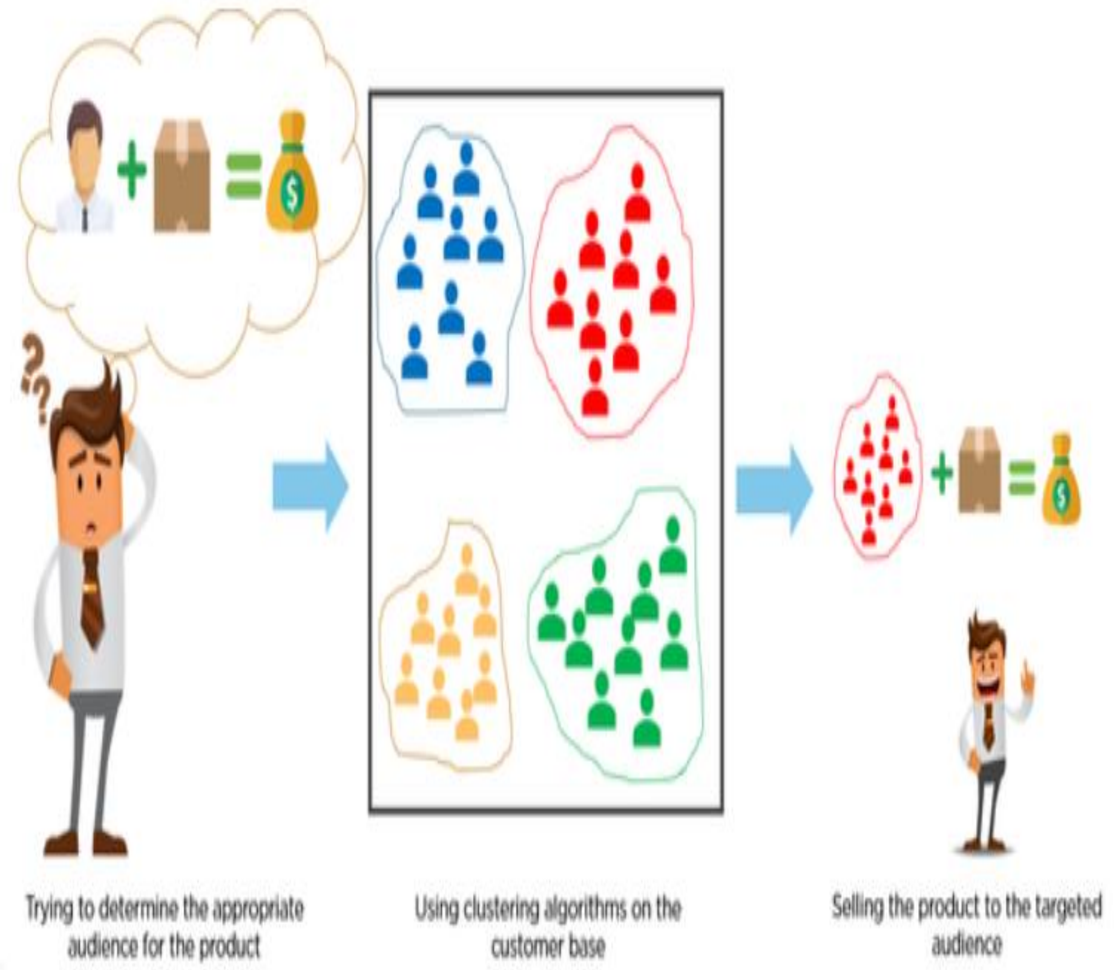
EXAMPLE 1: (IDENTIFYING TOPICS IN A SET OF BLOG POSTS)

- In a large collection of text data(for ex: set of blog posts), we might want to summarize it and find prevalent themes in it.
- Note that, we will NOT know beforehand what these topics are or how many topics there might be.



EXAMPLE 2: (SEGMENTING CUSTOMERS INTO GROUPS WITH SIMILAR PREFERENCES)

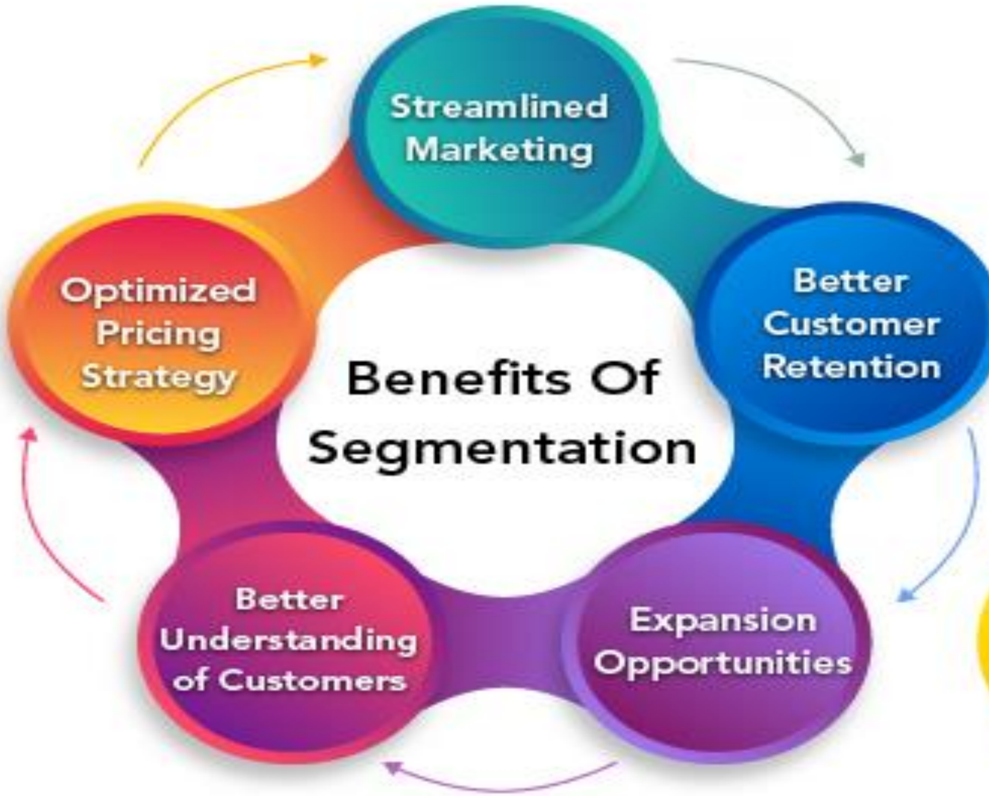
- Helps in determining thе appropriate audience for thе product and thereby the produc is sold to the targeted audience.
- For Ex.,

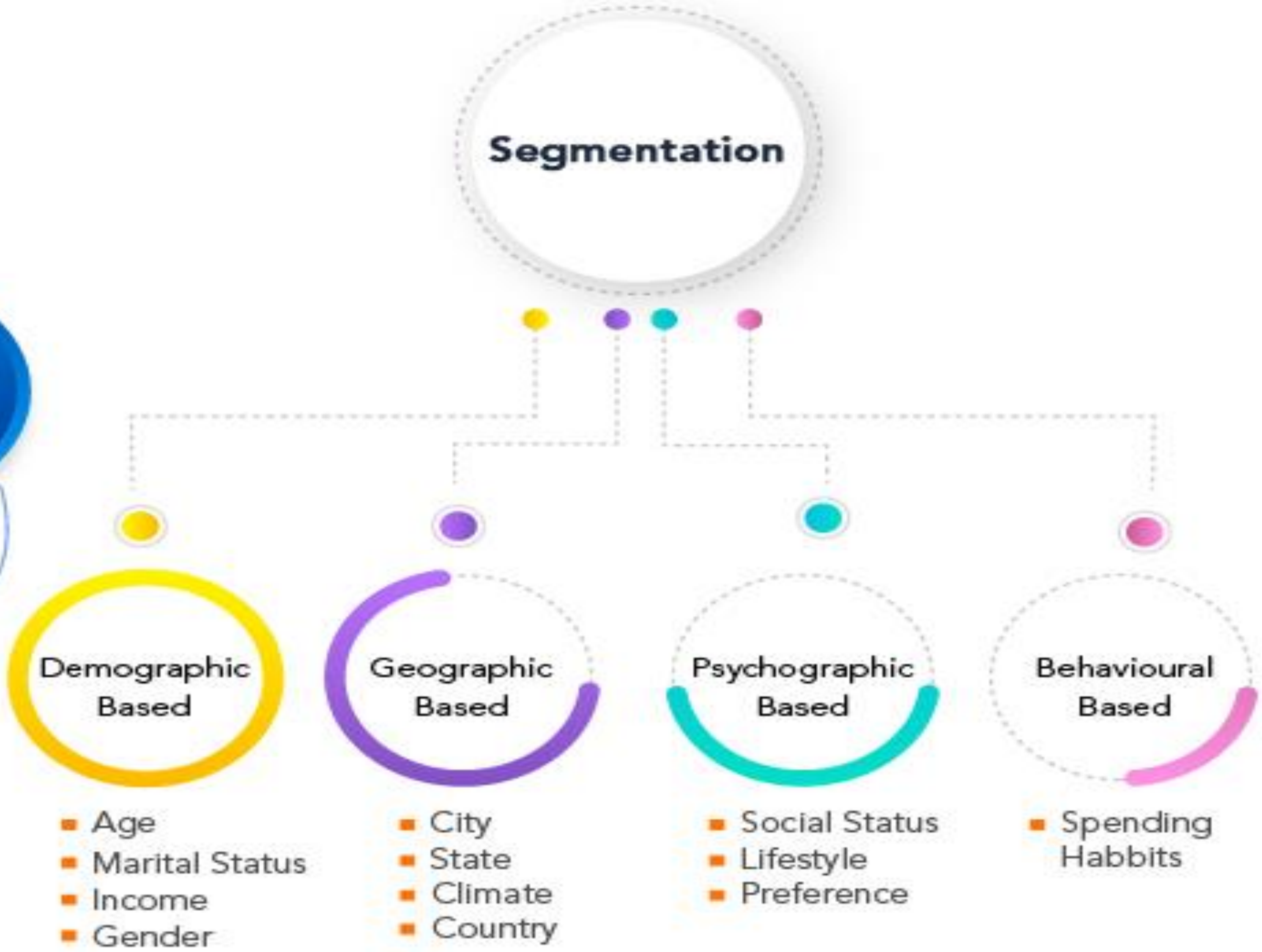


CUSTOMER SEGMENTATION IN VARIOUS BUSINESS DOMAINS

- Customer Segmentation in
  - Fashion Industry
  - Banking
  - Insurance
  - Pharmaceuticals
  - Travel Industry
  - Education

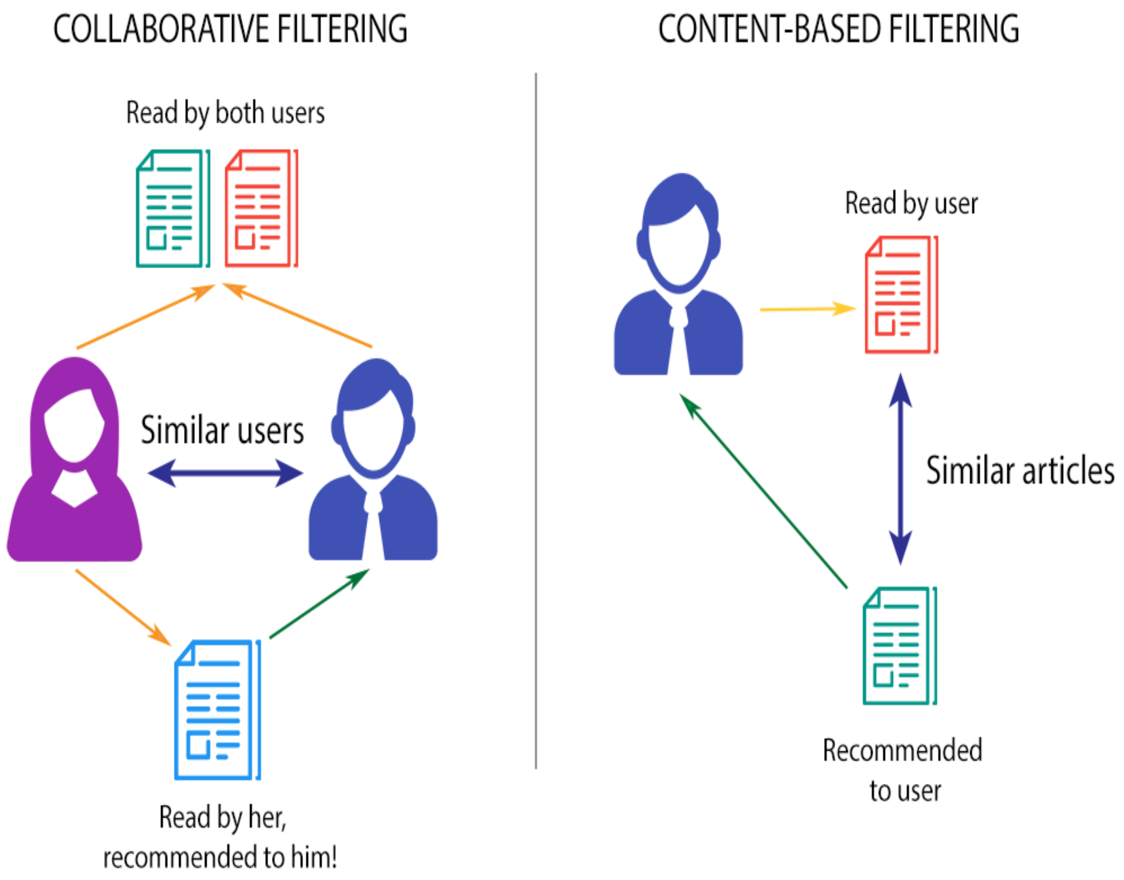
HOW DOES CUSTOMER SEGMENTATION BENEFIT THE BUSINESS





RECOMMENDER SYSTEMS

- We all have experienced the recommender systems online
- For Ex..Mobile recommendation
- Movie Recommendation



BEER-AND-DIAPERS CONNECTION

- The idea was to identify items that tended to be purchased together and place them together on store shelves.
- Doing so could boost sales by an additional percentage.
- If two products are kept next to each other on a shelf, they're more likely to be sold together.

NEED OF SEMI SUPERVISED LEARNING (SSL)

- Supervised learning is training a machine learning model using the labeled dataset.
- Organic labels are often available in data, but the process may involve a human expert who adds tags to raw data to show a model the target attributes (answers). In simple terms, a label is a description showing a model what it is expected to predict.
- Supervised learning has a few limitations. This process is

- ***slow*** (it requires human experts to manually label training examples one by one) and
- ***costly*** (a model should be trained on the large volumes of hand-labeled data to provide accurate predictions).

NEED OF SEMI SUPERVISED LEARNING (SSL)

Unsupervised learning, on the other hand, is when a model tries to mine hidden patterns, differences, and similarities in unlabeled data by itself, without human supervision.

Within this method, data points are grouped into clusters based on similarities.

The Disadvantage of Unsupervised Learning

- Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
- The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

NEED OF SEMI SUPERVISED LEARNING (SSL)

- Semi-supervised learning bridges supervised learning and unsupervised learning techniques to solve their key challenges.
- Here, an initial model is trained on a few labeled samples and then iteratively apply it to a greater number of unlabeled data.
- Unlike unsupervised learning, SSL works for a variety of problems from classification and regression to clustering and association.
- Unlike supervised learning, the method uses small amounts of labeled data and also large amounts of unlabeled data, which reduces expenses on manual annotation and cuts data preparation time.

EXAMPLE

Supervised learning

student is under the supervision of a teacher at both home and school

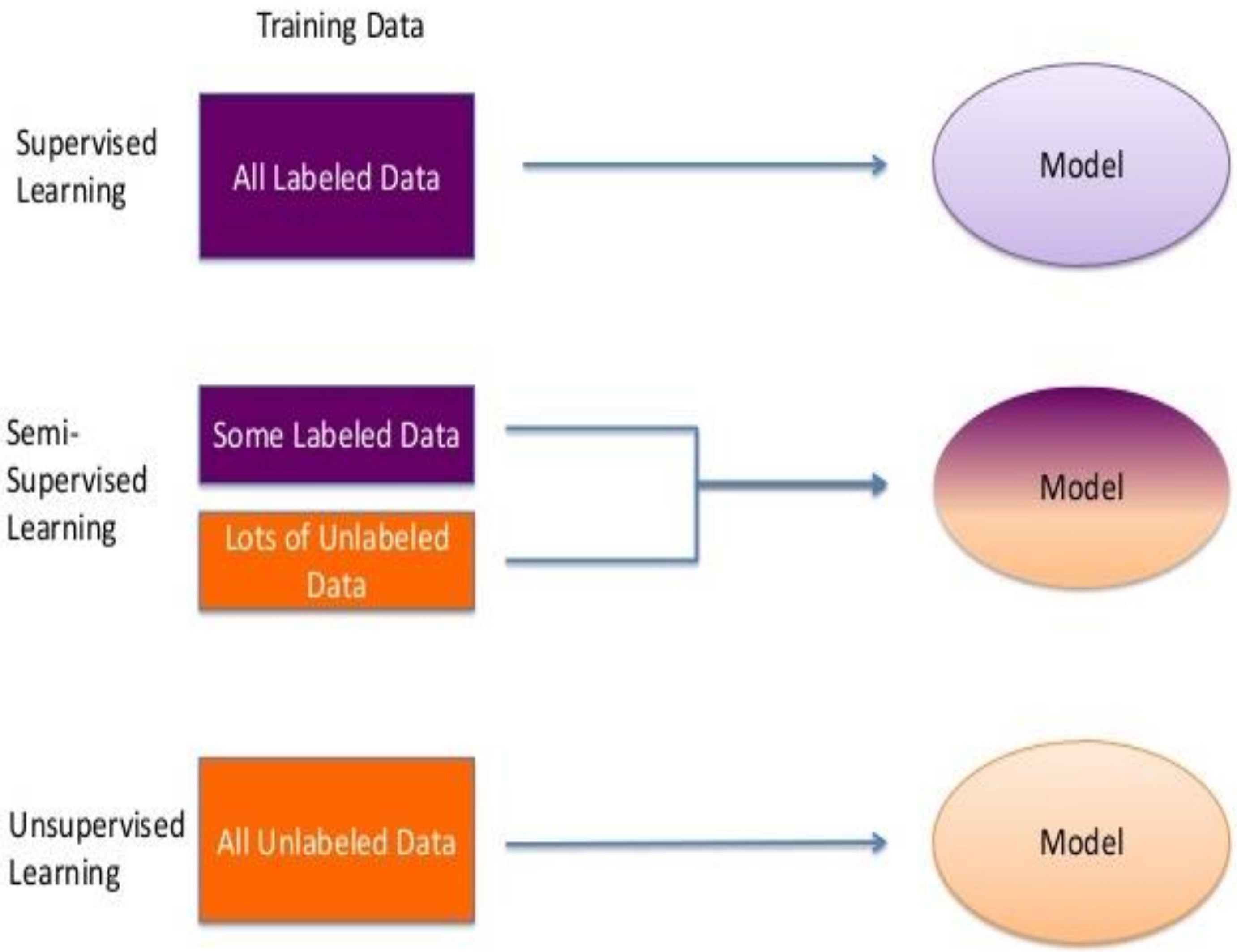
Unsupervised learning

student has to figure out a concept himself

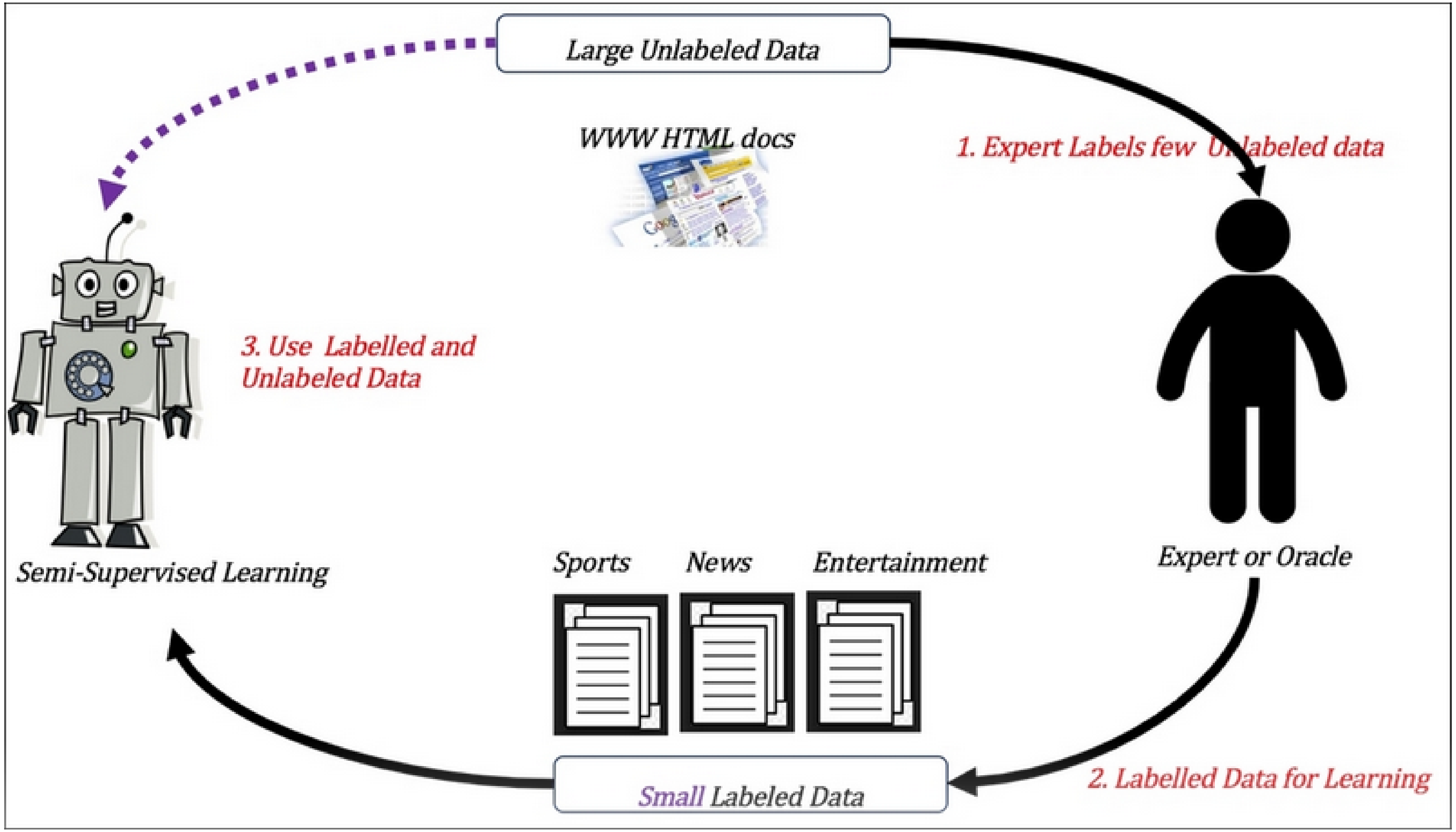
Semi-Supervised learning

teacher teaches a few concepts in class and gives questions as homework which are based on similar concepts.

3) SEMI SUPERVISED LEARNING



EXAMPLE-SEMI SUPERVISED LEARNING-DOCUMENT CLASSIFIER



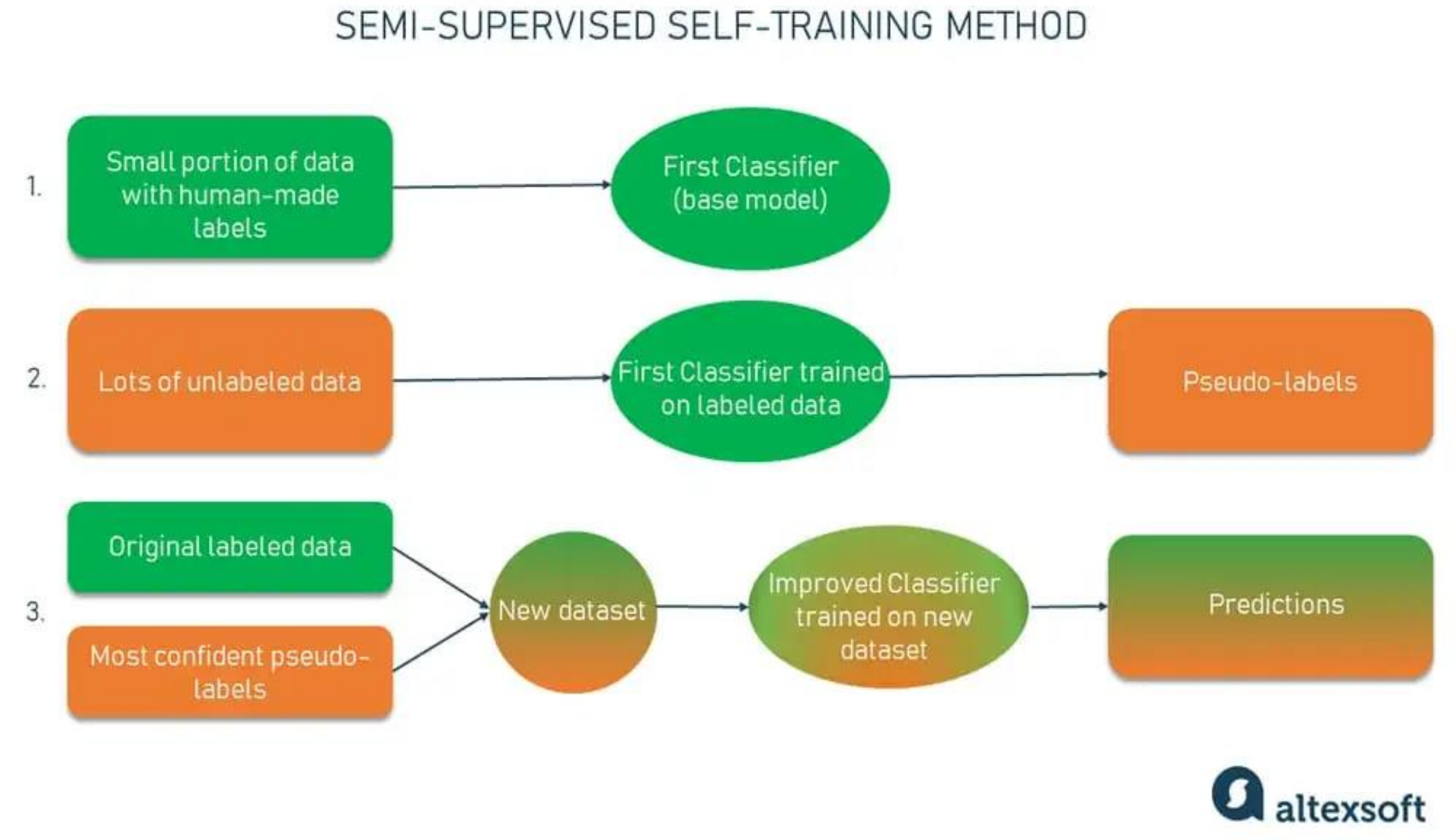
SEMI SUPERVISED LEARNING

- Semi-supervised learning is a branch of machine learning that combines supervised and unsupervised learning by using both labeled and unlabeled data to train artificial intelligence (AI) models for classification and regression tasks.
- It is a method that uses a small amount of labeled data and a large amount of unlabeled data to train a model.
- The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning. However, unlike supervised learning, the algorithm is trained on a dataset that contains both labeled and unlabeled data.
- Semi-supervised learning is particularly useful when there is a large amount of unlabeled data available, but it's too expensive or difficult to label all of it.

EXAMPLE_ SEMI-SUPERVISED LEARNING

Self-training

Self-training is the procedure in which you can take any supervised method for classification or regression and modify it to work in a semi-supervised manner, taking advantage of labeled and unlabeled data. The standard workflow is as follows.



Self-training Process:

- pick a small amount of labeled data, e.g., images showing cats and dogs with their respective tags, and you use this dataset to train a base model with the help of ordinary supervised methods.
- Then you apply the process known as **pseudo-labeling** —

  Take the partially trained model and use it to make predictions for the rest of the database which is yet unlabeled. The labels generated thereafter are called pseudo as they are produced based on the originally labeled data that has limitations (say, there may be an uneven representation of classes in the set resulting in bias — more dogs than cats).

- From this point, you take the most confident predictions made with your model (for example, you want the confidence of over 80 percent that a certain image shows a cat, not a dog). If any of the pseudo-labels exceed this confidence level, you add them to the labeled dataset and create a new, combined input to train an improved model.
- The process can go through several iterations (10 is often a standard amount) with more and more pseudo-labels being added every time. Provided the data is suitable for the process, the performance of the model will keep increasing at each iteration

CHALLENGES OF USING SEMI-SUPERVISED LEARNING

1. **Quality of unlabeled data**

The effectiveness of semi-supervised learning heavily depends on the quality and representativeness of the unlabeled data. If the unlabeled data is noisy or unrepresentative of the true data distribution, it can degrade model performance or even lead to incorrect conclusions.

For example, if you're using a dataset of product reviews for sentiment analysis, the unlabeled data might include reviews that are poorly written, contain sarcasm, or express neutral sentiment. If the model learns from these noisy unlabeled examples, it may misclassify similar reviews in the future, leading to lower accuracy and reliability in sentiment analysis predictions.

CHALLENGES OF USING SEMI-SUPERVISED LEARNING

2. Sensitivity to distribution shifts

Semi-supervised learning models may be more sensitive to distribution shifts between the labeled and unlabeled data. If the distribution of the unlabeled data differs significantly from the labeled data, the model's performance may suffer.

Example: Say that a model is trained on labeled images of cats and dogs from a dataset with high-quality photographs. However, the unlabeled data used for training contains images of cats and dogs captured from surveillance cameras with **low resolution** and poor lighting conditions.

If the distribution of images in the unlabeled data differs significantly from the labeled data, the model may struggle to generalize from the labeled to the unlabeled images, resulting in lower performance on real-world images with similar characteristics.

CHALLENGES OF USING SEMI-SUPERVISED LEARNING

3. Model Complexity

Some semi-supervised learning techniques, such as those based on generative models or adversarial training, can introduce additional complexity to the model architecture and training process.

Example:

Consider a semi-supervised learning approach that combines self-training with a language model pre-trained on a large corpus of text data. The model architecture may become increasingly complex due to the incorporation of multiple components. As the model complexity grows, it may become **more challenging to interpret, debug, and optimize**, leading to potential performance issues and increased computational resources required for training and inference.

4. Limited applicability

Semi-supervised learning may not be suitable for all types of tasks or datasets. It tends to be most effective when there is a sizable amount of unlabeled data available and when the underlying data distribution is relatively smooth and well-defined. This is why you should choose semi-supevised learning in those areas where its benefits outweigh the complexities.

SEMI-SUPERVISED LEARNING EXAMPLES

- **Speech recognition**
- Labeling audio is a very resource- and time-intensive task, so semi-supervised learning can be used to overcome the challenges and provide better performance.
- Facebook (now Meta) has successfully applied semi-supervised learning (namely the self-training method) to its speech recognition models and improved them.
- They started with the base model that was trained with 100 hours of human-annotated audio data. Then 500 hours of unlabeled speech data was added and self-training was used to increase the performance of the models. As far as the results, the word error rate (WER) decreased by 33.9 percent, which is a significant improvement.

SEMI-SUPERVISED LEARNING EXAMPLES

2. Web content classification

- With billions of websites presenting all sorts of content out there, classification would take a huge team of human resources to organize information on web pages by adding corresponding labels.
- The variations of semi-supervised learning are used to annotate web content and classify it accordingly to improve user experience.
- Many search engines, including Google, apply SSL to their **ranking component** to better understand human language and the relevance of candidate search results to queries.
- With SSL, Google Search finds content that is most relevant to a particular user query.

SEMI-SUPERVISED LEARNING EXAMPLES

3. Text document classification

Another example of when semi-supervised learning can be used successfully is in the **building of a text document classifier**. Here, the method is effective because it is really difficult for human annotators to read through multiple word-heavy texts to assign a basic label, like a type or genre.

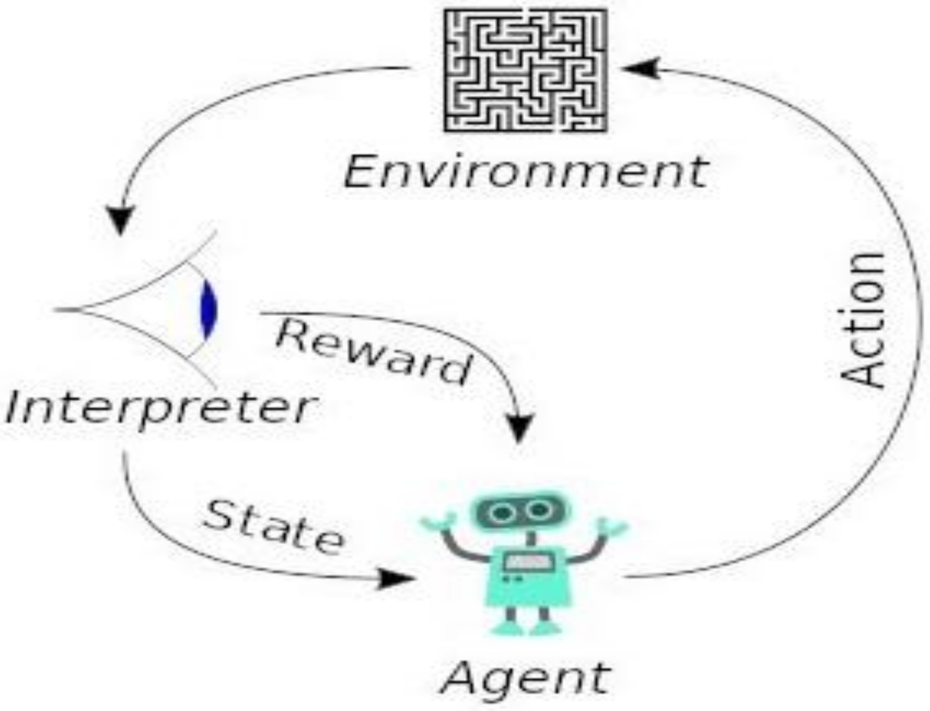
For example, a classifier can be built on top of deep learning neural networks like LSTM (long short-term memory) networks that are capable of finding long-term dependencies in data and retraining past information over time.

Usually, training a neural net requires lots of data with and without labels. A semi-supervised learning framework works just fine as you can train a base LSTM model on a few text examples with hand-labeled most relevant words and then apply it to a bigger number of unlabeled samples.

4. REINFORCEMENT LEARNING

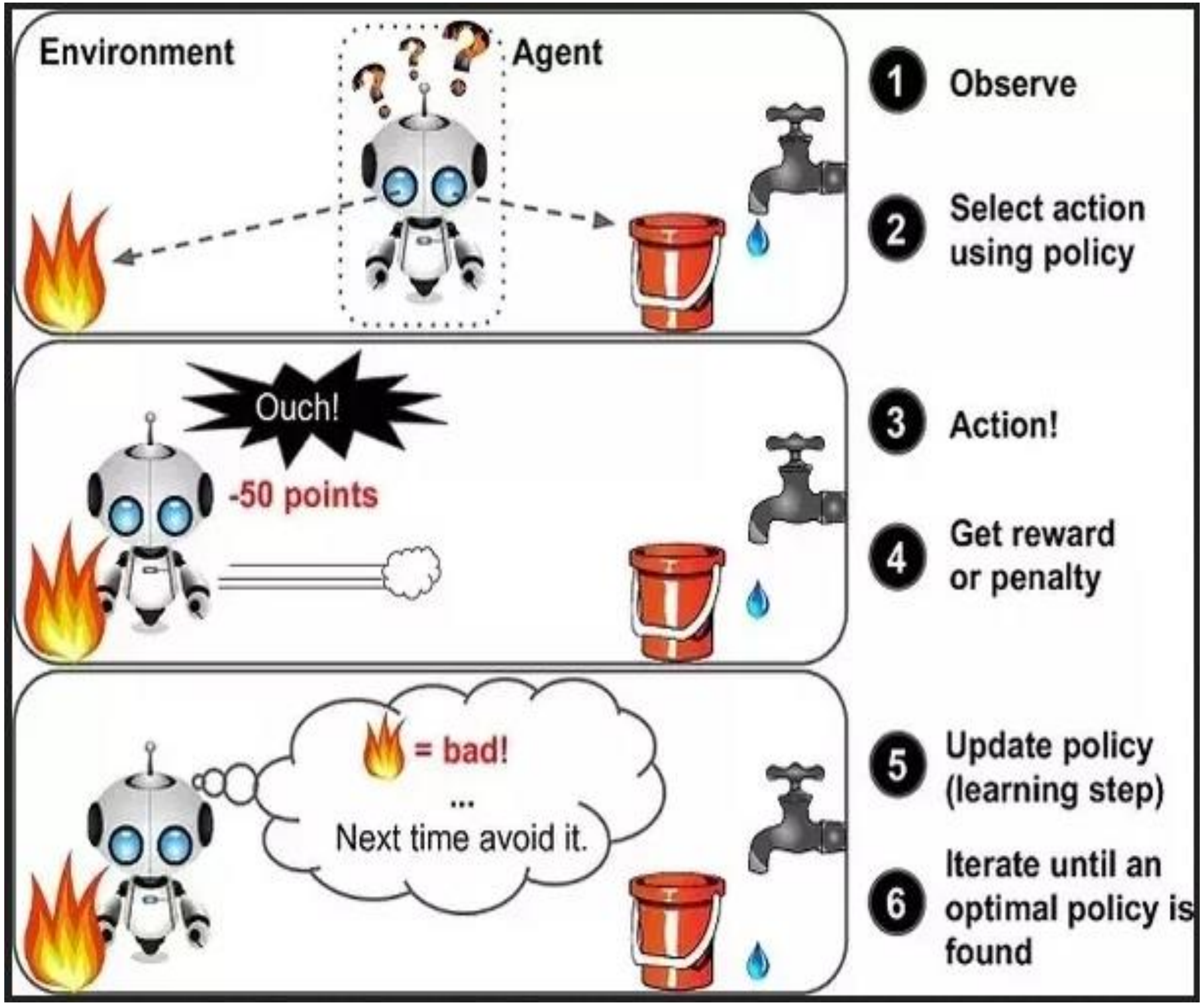
In **Reinforcement Learning (RL)**, agents are trained on a **reward** and **punishment** mechanism.

The **agent** is rewarded for correct moves and punished for the wrong ones. In doing so, the agent tries to minimize wrong moves and maximize the right ones.



There are 3 key terms in Reinforcement Learning

1. **State:** Describes the current situation. (e.g. Position in a maze)
2. **Action:** What the agent can do in its situation (e.g. Move right)
3. **Reward:** Feedback for whether a particular action in a given state was good or bad



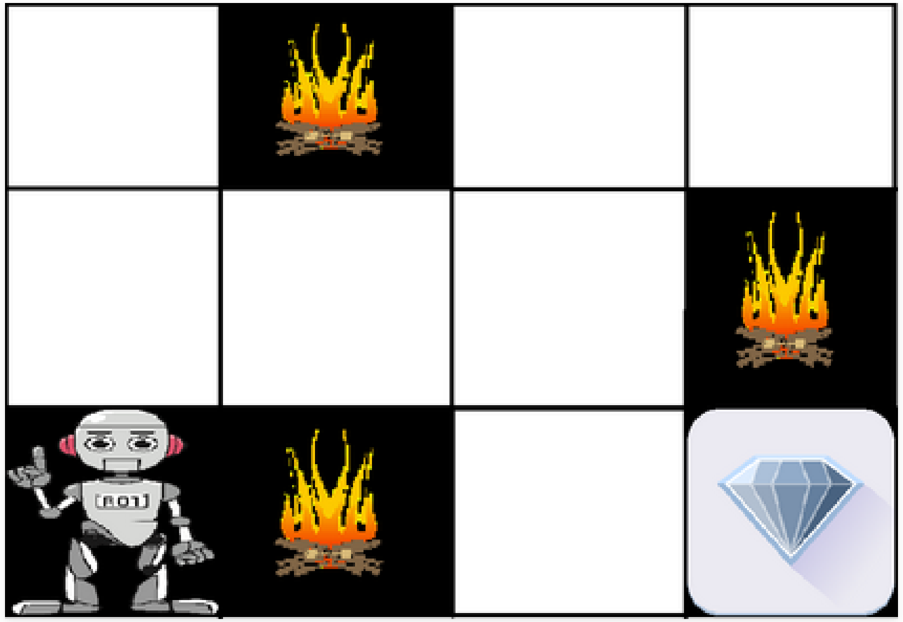
REINFORCEMENT LEARNING

- It is about taking suitable action to maximize reward in a particular situation.
- In the absence of a training dataset, it is bound to learn from its experience.

- For Ex., Self Driven Cars, Computers Learning Chess

Example: The problem is as follows:

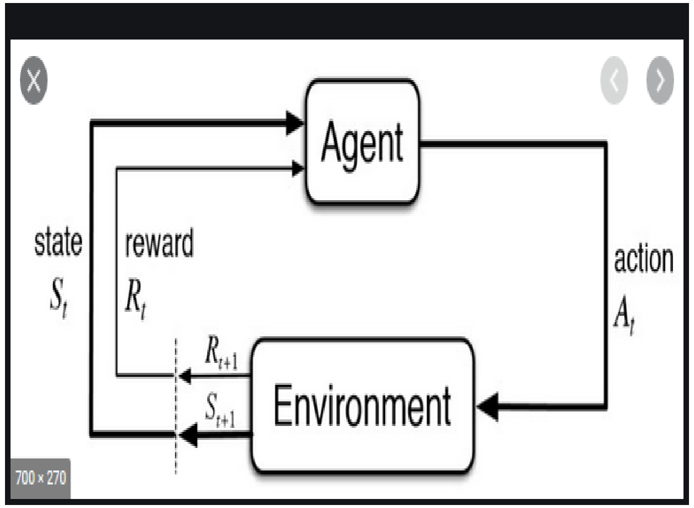
We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward.

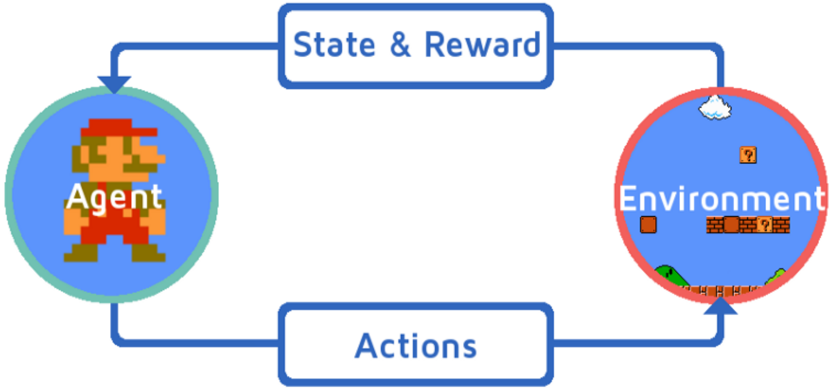


- The above image shows the robot, diamond, and fire.
- The goal of the robot is to
- **get the reward** that is the **diamond** and **avoid the hurdles** that are fire.
- The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles.
- Each right step will give the robot a reward and each wrong step will subtract the reward of the robot.
- The total reward will be calculated when it reaches the final reward that is the diamond.

Reinforcement learning

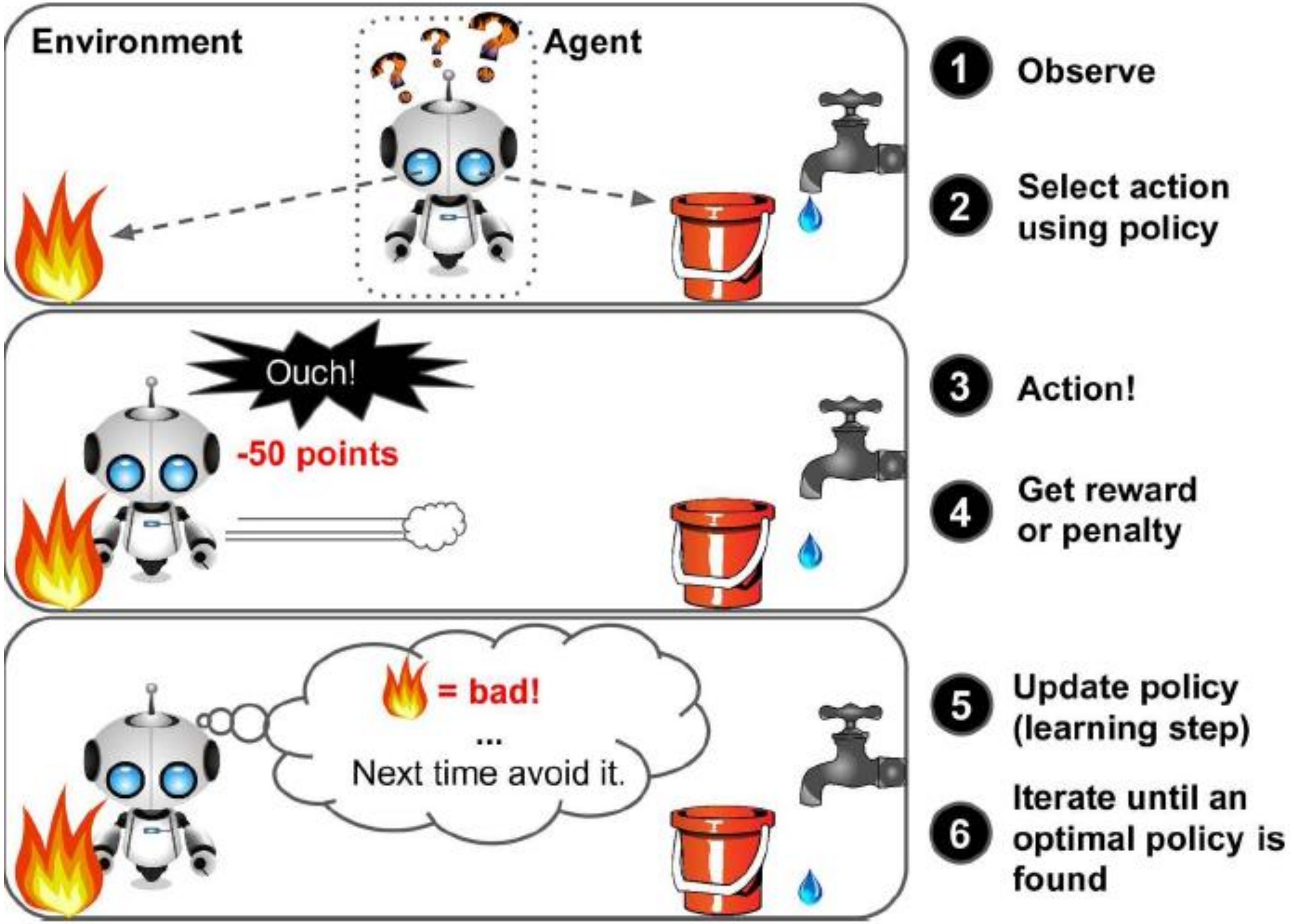
Reinforcement Learning is an area of machine learning concerned with how software agents ought to take optimal actions in an environment in order to achieve its goals.



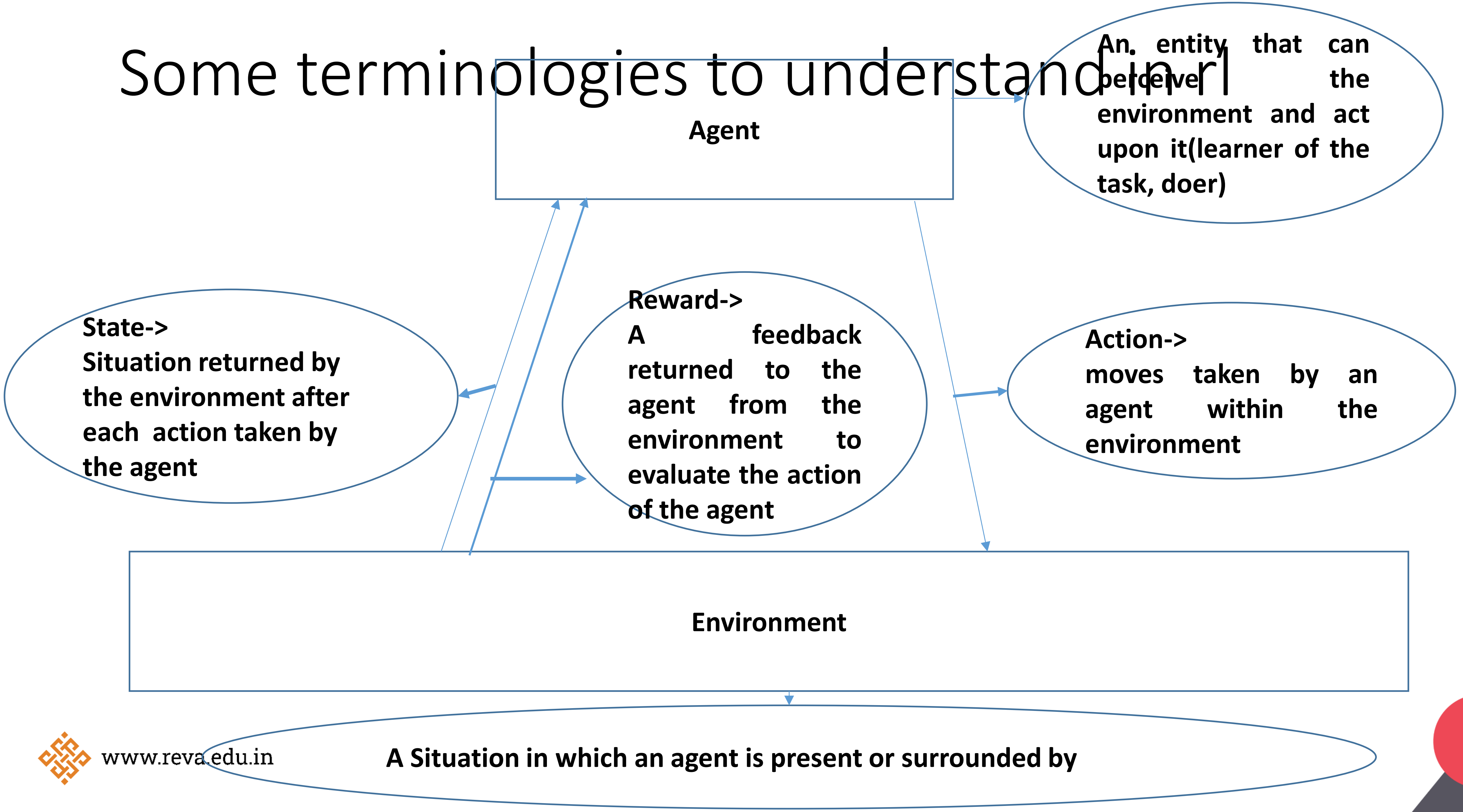


- *In Reinforcement Learning,* the learning system, called an *agent* observe the environment, select and perform actions, and get *rewards* in return (or *penalties* in the form of negative rewards).
- It must then learn by itself what is the best strategy, called a *policy*, to get the most reward over time.
- A policy defines what action the agent should choose when it is in a given situation.

Reinforcement learning



Some terminologies to understand in rl



Reinforcement learning

- It is about taking suitable action to maximize reward in a particular situation.
- In the absence of a training dataset, it is bound to learn from its experience.
- For Ex., Self Driven Cars, Computers Learning Chess

REINFORCEMENT LEARNING EXAMPLE

1) Self-Driven Cars

aspects to consider

- speed limits at various places
- drivable zones
- avoiding collisions

Autonomous driving tasks where reinforcement learning could be applied

- trajectory optimization
- motion planning
- dynamic pathing
- controller optimization and
- scenario-based learning policies for highways

REINFORCEMENT LEARNING EXAMPLE

2) **Trading and finance**

- Supervised **time series** models can be used for predicting future sales as well as predicting **stock prices**.
- These models don't determine the action to take at a particular stock price.
- An RL agent can decide on such a task; whether to hold, buy, or sell.

Ex: IBM

Financial trades. It computes the reward function based on the loss or profit of every financial transaction.

REINFORCEMENT LEARNING EXAMPLE

3) **NLP (Natural Language Processing)**

In NLP, RL can be used in

- **Text summarization**
- **Question answering** and
- **Machine translation**

REINFORCEMENT LEARNING EXAMPLE

4) Engineering

Facebook has developed an **open-source reinforcement learning platform**— Horizon to optimize large-scale production systems.

Facebook has used Horizon internally:

1. To personalize suggestions
2. Deliver more meaningful notifications to users
3. Optimize video streaming quality

REINFORCEMENT LEARNING EXAMPLE

5. **News recommendation**

User preferences can change frequently, therefore **recommending news** to users based on reviews and likes could become obsolete quickly.

With reinforcement learning, the RL system can track the reader's return behaviors.

Construction of such a system would involve obtaining

1) news features : ex: content, headline, and publisher

2) reader features: how the reader interacts with the content

    e.g clicks and shares

1) context features: timing and freshness of the news

REINFORCEMENT LEARNING EXAMPLE

6. Gaming

Ex: **AlphaGo Zero**

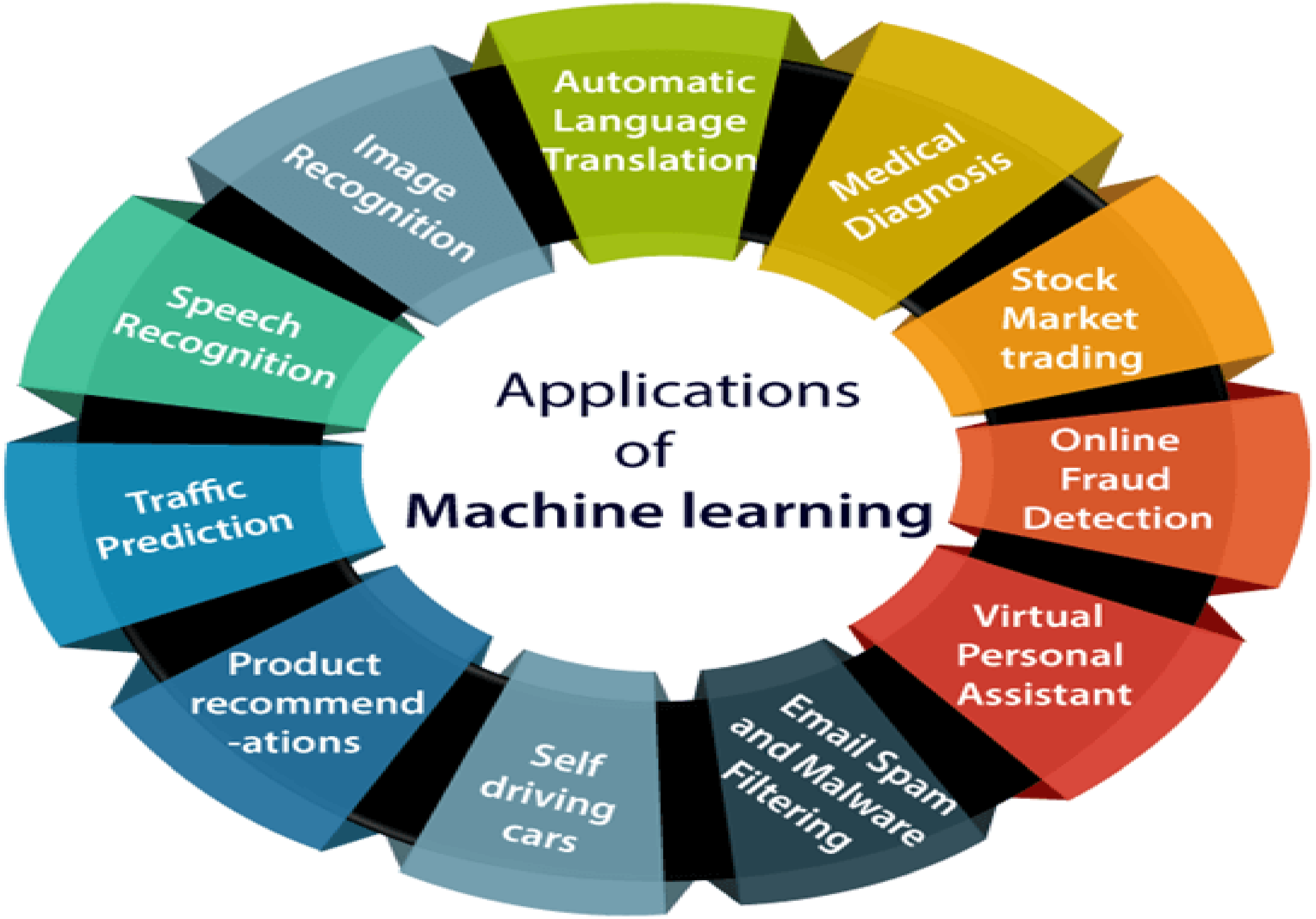
- able to learn the game of Go from scratch.
- It learned by playing against itself.
- After 40 days of self-training, has defeated world number one **Ke Jie**

REINFORCEMENT LEARNING EXAMPLE

7.

Design a traffic light controller to solve the congestion problem

Applications of machine learning



Speech recognition system

- **Speech recognition** is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format.
- Ex., Alexa, Cortana, Google Assistant and Siri are changing the way people interact with their devices, homes, cars, and jobs

Speech to text conversion

- **Speech to Text** conversion is the process of converting spoken words into written texts

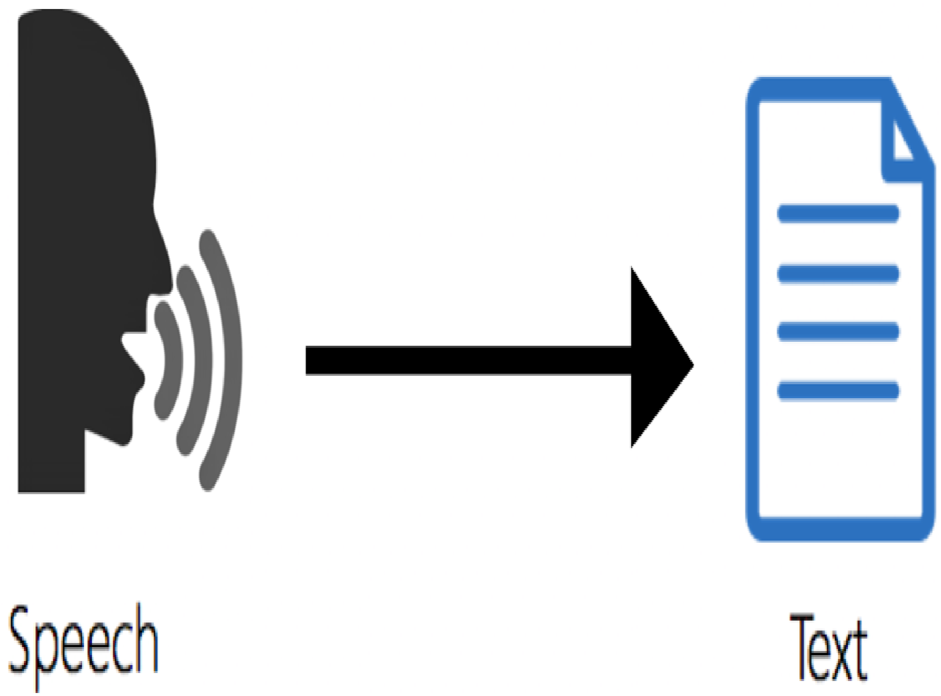
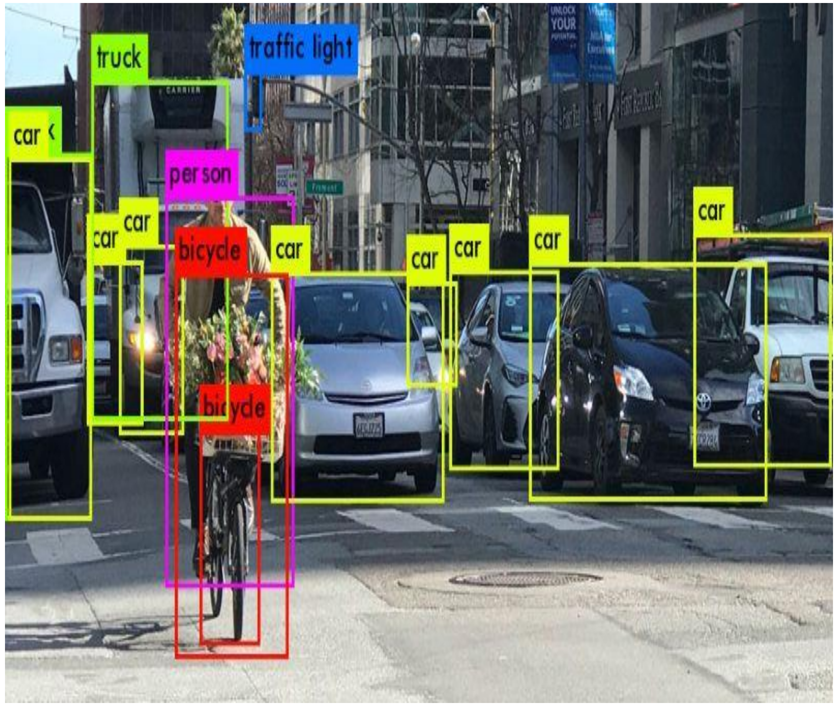


Image recognition

Major applications of image recognition are

- face recognition,
- Security & Surveillance,
- visual geolocation,
- object recognition,
- gesture recognition,
- code recognition,
- industrial automation,
- image analysis in medical and driver assistance.



Automatic language translation

- **Automatic language translation** is the use of a computer program to **translate** input text from one national **language**

Machine learning in stock market investment

- The primary objective of investing is to ensure that every person is able to meet his or her future financial objectives.
- Rise in inflation makes it inadequate for individuals to simply earn and save some part of their incomes. To meet the price increases due to inflation, investments become important.
- The stock market is one of the oldest and most popular investment avenues due to several benefits of investing in stocks.

Machine learning in stock market investment

- Using features like the latest announcements about an organization, their quarterly revenue results, etc.,
- Machine learning techniques have the potential to unearth patterns and insights we didn't see before
- These can be used to make unerringly accurate predictions.

Colgate’s Smart Toothbrush

- Colgate’s smart toothbrush has been trained to get paired with one’s smartphone.
- Colgate’s new Plaqless Pro smart toothbrush, which is an electric type, has tiny sensors embedded that can detect the plaque build up inside the user’s mouth before one starts brushing.

Colgate's Smart Toothbrush

- One needs to pair their toothbrush to smartphones via Bluetooth, and the Colgate app analyses the mouth and the data for the user.
- This analysing and deciphering will be unique to each user's mouth.
- The data collected and analysed will give details about whether the user's brushing technique is a viable one and whether the individual has terrible brushing techniques, and all this analysis will be delivered in real-time on the Colgate application.
- This app will also tell the user if they have missed a spot and even the history of their brushing techniques along with customised oral care tips.

Use case: How uber is driven by machine learning

- Uber gives about 1 million rides per day and 14,000 rides per minute and adds about 50,000 drivers per month.
- Giving customers exactly what they want at a reasonable cost is one of the important reasons for Uber’s success.
- Uber’s machine learning algorithms play a crucial role in helping the company predict customer needs.

How uber is driven by machine learning

Bridging the supply-demand gap

- Based on historical data, Uber predicts the time and areas of demand.
- The system uses these predictions to alert drivers of the areas with upcoming demand.
- This way, Uber makes sure that there are enough cabs in the predicted areas of demand and bridges the supply-demand gap.
- Demand prediction systems enable the app to slightly increase the prices during peak hours, eventually increasing profitability.

How uber is driven by machine learning

Reduction in ETA(Expected Arrival Time)

- The time wasted in road traffic is one of the most frustrating problems in the urbanized areas.
- This gets worse when cabs take longer to reach the pickup point. But, Uber's machine learning algorithms have a solution for this issue too.
- By predicting demand and keeping cabs ready, Uber reduces the expected time arrival (ETA) when a customer makes a booking.
- By reducing a lot of wait time, Uber always makes the customer experience better.
- The perfect blend of customer satisfaction, loyalty programs, and referral bonuses has resulted in a massive expansion in the customer base just through word of mouth.

How uber is driven by machine learning

Route optimization

- Conventional ride-hailing systems require the driver to make assumption-based route choices.
- This method is not reliable because the travel duration through the same route might change based on traffic jams, weather conditions, and road maintenance schedules.
- But Uber’s machine learning system updates the app with the conditions in every route and suggests the fastest route to the driver.
- This way, Uber helps its drivers avoid congestion and enables faster rides.
- Besides making the customers happy, faster rides also enable drivers to get additional time to take on more rides.

How uber is driven by machine learning

AI-based one-click chat

- Riders tend to message drivers while they wait for the cab.
- Most of the time, riders do this to check the status when they see the cab barely moving in the app.
- It is difficult for drivers to type a reply while driving.
- So Uber came up with an artificial intelligence-based concept called the “one-click chat.”
- The one-click chat leverages natural language processing and machine learning techniques to predict responses to common messages.
- This way, drivers can easily respond to the messages by just clicking on one of the suggested replies.

How uber is driven by machine learning

Uber Pool

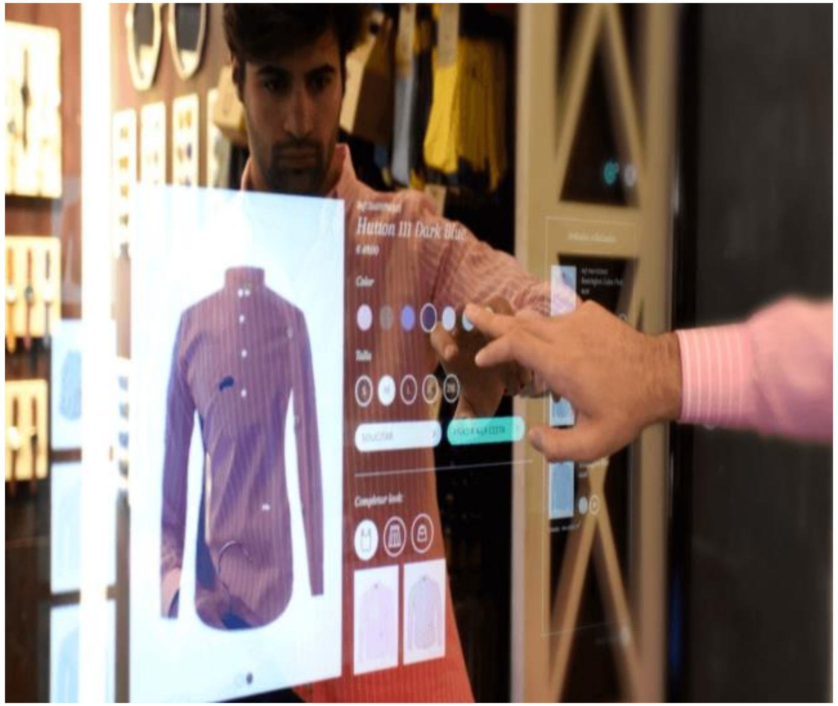
- During rush hours, it is difficult to make individual cabs available for everyone.
- But ridesharing solves this problem by matching the riders heading in the same direction.
- Also, the pooling feature makes the rides more economical by reducing the fares by 25 percent to 40 percent.
- Machine learning algorithms decide which rider to drop first based on the data gathered from maps.
- Also, the app uses historical data and patterns to understand peak hours and surge prices accordingly.

AI in Fashion with Smart Mirror

- The AI smart mirror is installed in the changing room of retail stores with touch screen glasses that relay information on whether or not a person is inside
- They will also help to get information about the item the customer has brought into the store.

AI in Fast Fashion with Smart Mirror

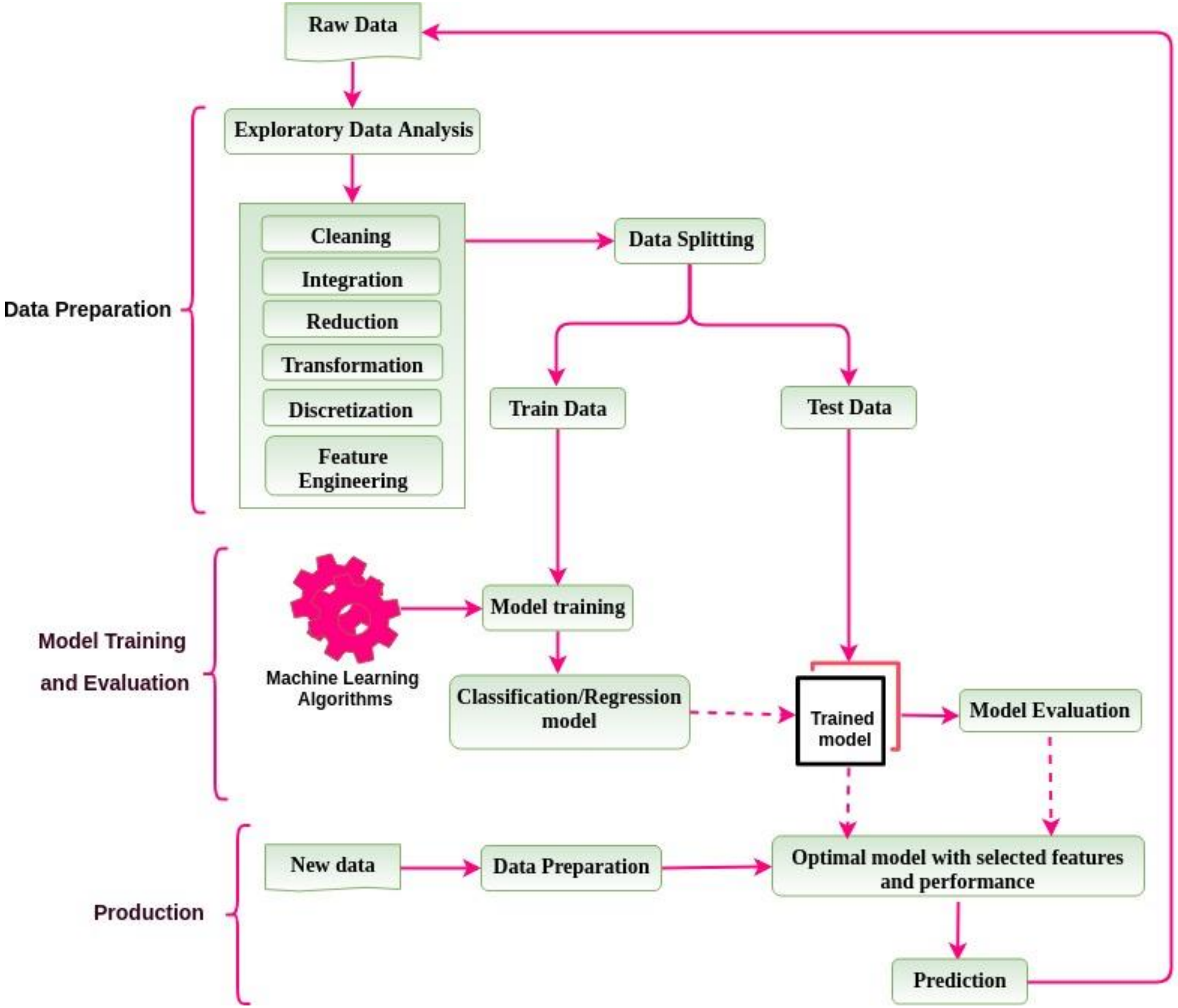
- For such smart mirrors, clothing racks are RFID enabled
- use gyro-sensors and Bluetooth low-energy chips allowing the articles selected by shoppers automatically show up in the Smart Mirror.



How banks are using machine learning to detect Fraud detection

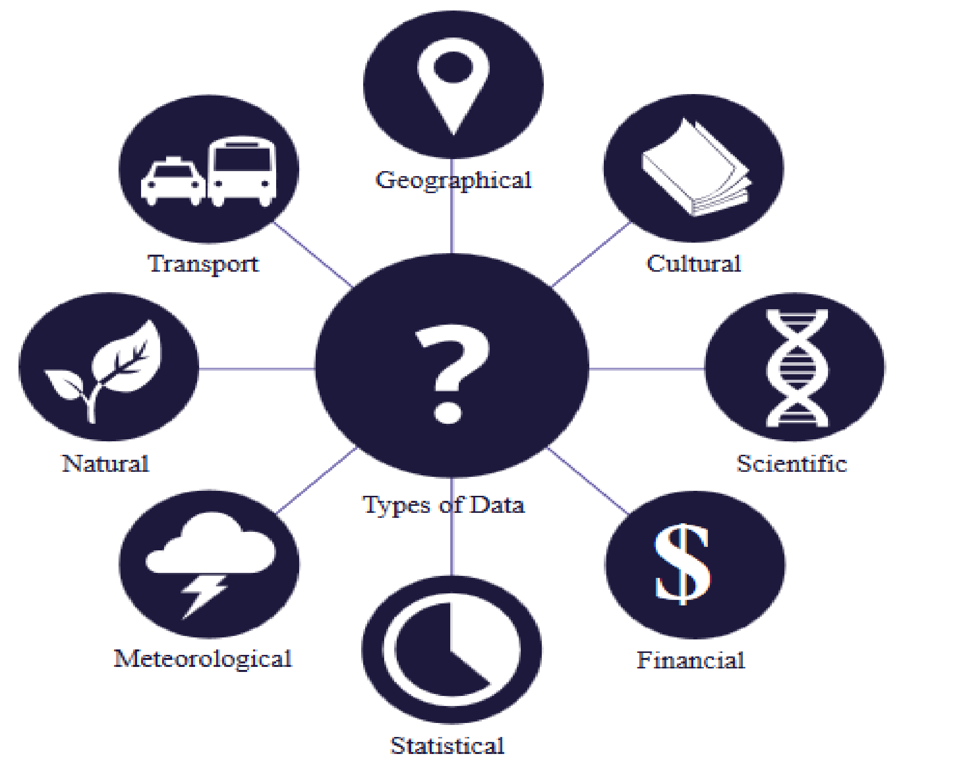
- It is also very common to use methods to tweak the data distribution beforehand (supervised learning), thus achieving a better balance between classes (fraud, not fraud)
- a problem that represents a serious challenge given that only one case out of tens of thousands is raised.
- Techniques are used the identification of anomalies or other types of pattern recognition models (unsupervised learning)

MACHINE LEARNING WORKFLOW



What is data all about?

- **Data** are individual pieces of factual information recorded and used for the purpose of analysis.
- The data can be of facts of various aspects in various Domains



how so much of data???(sources of data)

Various online Transactions done.Ex-Paying Bills, shopping online,Selling Online

Under Academics-Innumerable courses offered online

1.7 Million Pictures are uploaded per minute on Instagram
3,47,222 tweets/min
204000000
300 hours of videos uploaded per minute

Sources Of Data

Data From Facebook

IoT Generates 2.5 Quantillion bytes per day

In Smart Phones coz of Apps usage, games, chat messages

What is quality of data??

- Data is said to be of Quality if it satisfies the requirements of the intended use. Factors contributing to data quality are:
- Accuracy
- Completeness
- Consistency
- Timeliness
- Believability
- Interpretability

WHY DATA PRE PROCESSING?

Data in the real world is dirty

– **incomplete**: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data

• e.g., occupation=“ ”

– **noisy**: containing errors or outliers

• e.g., Salary=“-10”

– **inconsistent**: containing discrepancies in codes or names

• e.g., Age=“42” Birthday=“03/07/1997”

• e.g., Was rating “1,2,3”, now rating “A, B, C”

• e.g., discrepancy between duplicate records

WHY IS DATA PREPROCESSING IMPORTANT?

No quality data, no quality results!

– Quality decisions must be based on quality data

• e.g., duplicate or missing data may cause incorrect or even misleading statistics.

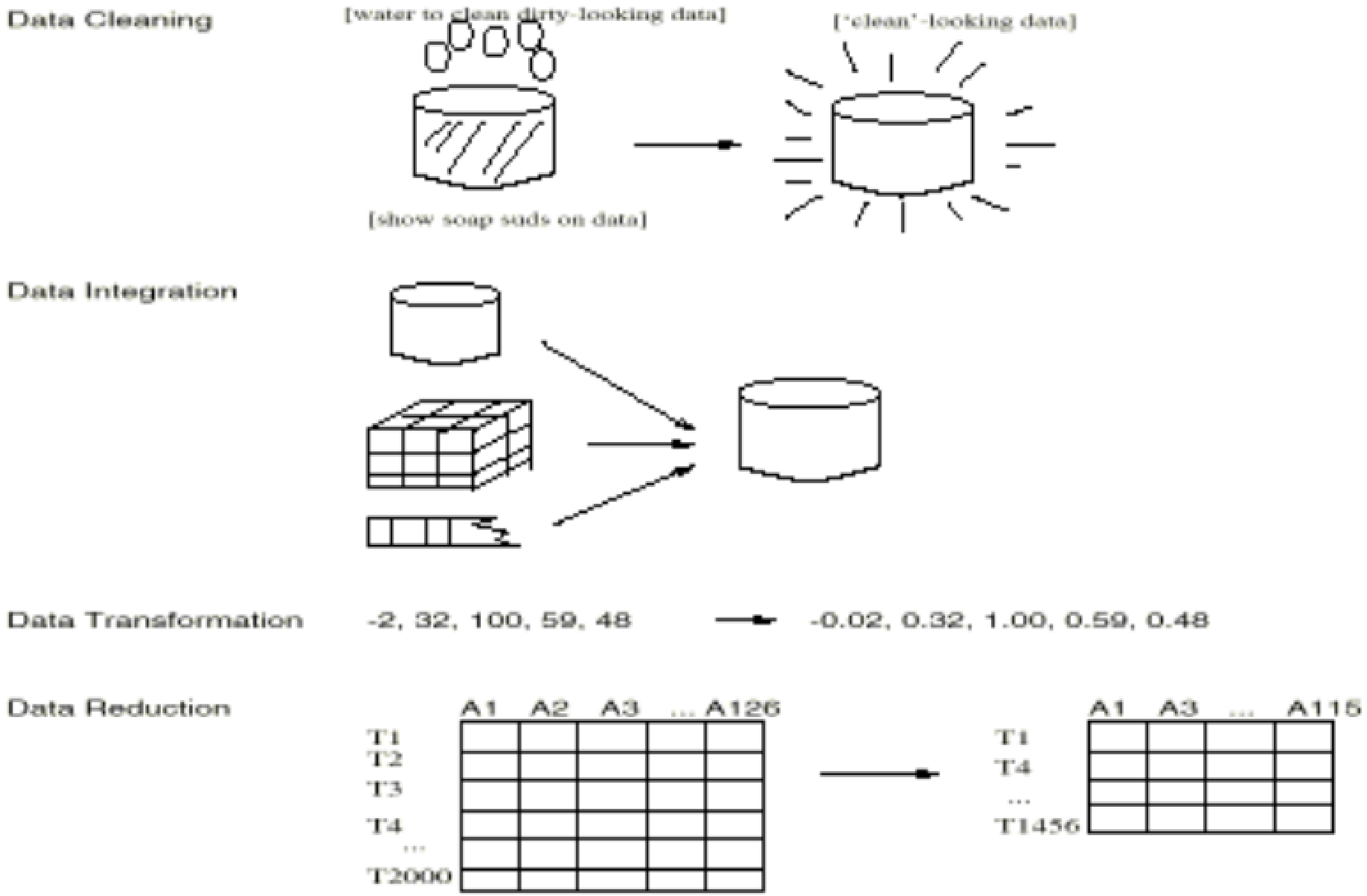
– ML needs consistent integration of quality data

• Data extraction, cleaning, and transformation comprises the majority of the work of building ML model.

MAJOR TASKS IN DATA PREPROCESSING

- **Data cleaning**
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
  - Integration of multiple databases, data cubes, or files
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

FORMS OF DATA PRE PROCESSING



1) DATA CLEANING

Importance

– "Data cleaning is one of the three biggest problems in data warehousing"

—Ralph Kimball

– "Data cleaning is the number one problem in data warehousing"- DCI survey

- **Data cleaning tasks**

– Fill in missing values

– Identify outliers and smooth out noisy data

– Correct inconsistent data

– Resolve redundancy caused by data integration

Handling Missing data

- Data in the real world is rarely clean and homogeneous.
- In particular, many interesting datasets will have some amount of data missing.
- To make matters even more complicated, different data sources may indicate missing data in different ways.
- **Some general considerations for missing data, discuss how Pandas chooses to represent it, and demonstrate some built-in Pandas tools for handling missing data in Python.**
- **Refer to missing data in general as null, NaN, or NA values**

NaN: Missing numerical data

```python
# Load data
data = pd.DataFrame({
'age': [25, np.nan, 30, 45, np.nan],
'salary': [50000, 60000, np.nan, 65000, 70000],
'city': ['New York', 'Los Angeles', 'New York', 'San Francisco', np.nan],
'target': [1, 0, 1, 0, 1]
})
```

```
Original Data:
    age   salary           city  target
0  25.0  50000.0       New York       1
1   NaN  60000.0    Los Angeles       0
2  30.0      NaN       New York       1
3  45.0  65000.0  San Francisco       0
4   NaN  70000.0            NaN       1
```

Identifying missing values

```
# Display missing values
print("Missing values:")
print(data.isnull().sum())
```

```
Missing values:
age       2
salary    1
city      1
target    0
dtype: int64
```

1. Deleting the rows of missing values

```
# removing all  rows with null values
data1 = data # creating a copy of the data
data1 = data1.dropna()
print('after removing rows')
print(data1)
```

```
after removing rows
    age   salary           city  target
0  25.0  50000.0       New York       1
3  45.0  65000.0  San Francisco       0
```

2. Replacing the missing values of numerical attributes with mean

```
from sklearn.impute import SimpleImputer
# For numerical features, use mean imputation
num_features = ['age', 'salary'] # Creating a list of numerical features
imputer_num = SimpleImputer(strategy='mean') # creating an instance of simple imput
data[num_features] = imputer_num.fit_transform(data[num_features]) # fitting to the
data
```

	age	salary	city	target
0	25.000000	50000.0	New York	1
1	33.333333	60000.0	Los Angeles	0
2	30.000000	61250.0	New York	1
3	45.000000	65000.0	San Francisco	0
4	33.333333	70000.0	NaN	1

3. Replacing the missing values of categorical attributes with the most frequent values

```
# For categorical features, use the most frequent value imputation
cat_features = ['city'] #creating a list of categorical attributes
imputer_cat = SimpleImputer(strategy='most_frequent') # creating a instance of simp
data[cat_features] = imputer_cat.fit_transform(data[cat_features]) # fitting and tr
data
```

	age	salary	city	target
0	25.000000	50000.0	New York	1
1	33.333333	60000.0	Los Angeles	0
2	30.000000	61250.0	New York	1
3	45.000000	65000.0	San Francisco	0
4	33.333333	70000.0	New York	1

4. Label encoding of categorical values

Label encoding assigns a unique integer to each distinct category in a feature.

```
from sklearn.preprocessing import StandardScaler, LabelEncoder
# Encoding categorical features
cat_features = ['city']
label_encoders = {}
for col in cat_features:
    le = LabelEncoder() # creating an instance of labelencoder
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le
```

```
data
```

	age	salary	city	target
0	25.000000	50000.0	1	1
1	33.333333	60000.0	0	0
2	30.000000	61250.0	1	1
3	45.000000	65000.0	2	0
4	33.333333	70000.0	1	1

Data integration

- Data Integration is the merging of data from multiple data sources such as databases, CSV files, APIs, or real-time data streams.
- **Techniques**: Use appropriate techniques to extract data from various sources.

✓ **SQL Queries**: For relational databases.

✓ **Web Scraping**: For data from websites.

✓ **API Calls**: For data from web services.

✓ **File Reading**: For CSV, Excel, or JSON files.

- Data mapping and consolidation using schema alignment, merging and aggregation
- Store the data in file format as required

Data reduction

- Data Reduction is the process of obtaining reduced representation of the data set that is much smaller in volume but still contain critical information.
- Data Reduction is a technique of obtaining a reduced representation of the dataset, that is much smaller in volume, yet closely maintains the integrity of the original data and produce the same analytical results.

Why Data reduction

When the data set is likely to be huge, complex data analysis on huge amounts of data can take a long time, making such analysis impractical or infeasible.

DATA REDUCTION STRATEGIES

1) Dimensionality reduction, e.g., remove unimportant attributes

    a. Wavelet transforms

    b. Principal Components Analysis (PCA)

    c. Feature subset selection, feature creation or Attribute subset selection

2. Numerosity reduction (some simply call it: Data Reduction)

    a. Regression and Log-Linear Models

    b. Histograms, clustering, sampling

    c. Data cube aggregation

DATA TRANSFORMATION

- A function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values

DATA TRANSFORMATION METHODS

1. Smoothing, which works to remove noise from the data. Techniques include binning, regression, and clustering.
2. Attribute construction (or feature construction), where new attributes are constructed and added from the given set of attributes to help the mining process.
3. Aggregation, where summary or aggregation operations are applied to the data.

For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for data analysis at multiple abstraction levels.

4. Normalization, where the attribute data are scaled so as to fall within a smaller range, such as−1.0 to 1.0, or 0.0 to 1.0.

3 methods

Min-max normalization

Z-score normalization

Normalization by decimal scaling

NORMALIZATION

a. **Min-max normalization**: to [new_min$_A$, new_max$_A$]

Min-max normalization performs a linear transformation on the original data. Suppose that min_A and max_A are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, v_i, of A to v_i' in the range $[new\_min_A, new\_max_A]$ by computing

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Suppose that the minimum and maximum values for the attribute income are $12,000 and $98,000, respectively.

We would like to map income to the range [0.0,1.0].

- Then $73,600 is mapped to $\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$

NORMALIZATION

b. Z-score normalization (μ: mean, σ: standard deviation):

In **z-score normalization** (or *zero-mean normalization*), the values for an attribute, A, are normalized based on the mean (i.e., average) and standard deviation of A. A value, v_i, of A is normalized to v_i' by computing

$$v' = \frac{v - \mu_A}{\sigma_A}$$

where $\bar{A}$ and σ_A are the mean and standard deviation, respectively, of attribute A.

Example: Suppose that the mean and standard deviation of the values for the attribute income are \$54,000 and \$16,000, respectively. With z-score normalization, a value of \$73,600 for income is transformed to

$$\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$$

3) Normalization by decimal scaling

Normalization by decimal scaling normalizes by moving the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A.

A value, v_i, of A is normalized to v'_i by computing

$$v' = \frac{v}{10^j}$$

Example: Suppose that the recorded values of A range from −986 to 917.

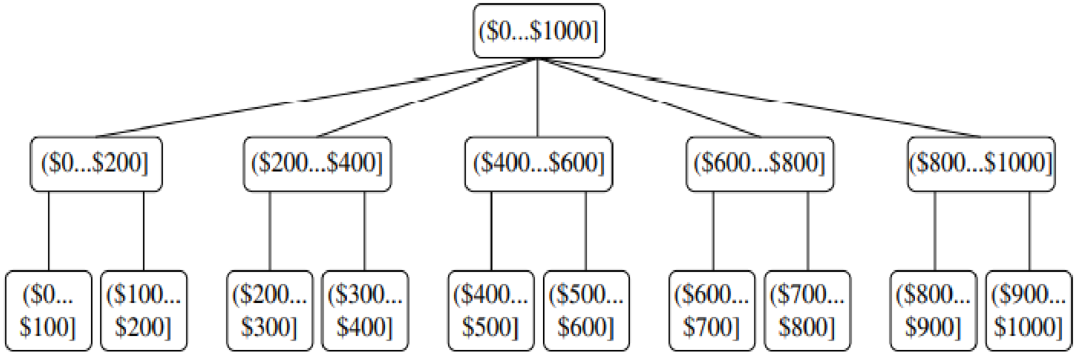
The maximum absolute value of A is 986. To normalize by decimal scaling, we therefore divide each value by 1000 (i.e., j = 3)

so that −986 normalizes to −0.986 and 917 normalizes to 0.917.

DT METHODS

5. Discretization

- where the raw values of a numeric attribute (e.g . age) are replaced by
  - interval labels (e.g., 0–10, 11–20, etc.) or
  - conceptual labels (e.g. youth, adult, senior).
- The labels, in turn, can be recursively organized into higher-level concepts, resulting in a concept hierarchy for the numeric attribute.
- Figure shows a concept hierarchy for the attribute price. More than one concept hierarchy can be defined for the same attribute to accommodate the needs of various users.



A concept hierarchy for the attribute *price*, where an interval ($\$X \ldots \$Y$] denotes the range from $\$X$ (exclusive) to $\$Y$ (inclusive).

Applying Data Transformations

- Preprocessing methods like the scalers are usually applied before applying a supervised machine learning algorithm.
- Apply the kernel SVM (SVC) to the cancer dataset, and use MinMaxScaler /StandardScalar for preprocessing the data.

Example:

5. Scaling the numerical attribute values using standard scalar

```
#Feature scaling
scaler = StandardScaler() #creating a instance of standard scalar
num_features = ['age','salary']
data[num_features] = scaler.fit_transform(data[num_features]) # Fitting the data
data
```

TRAINING AND CHOOSING PREDICTIVE MODELS

ML Model Training?

Definition:
Model training is the process where a machine learning algorithm learns patterns from input data (features) and their corresponding outputs (labels), so that it can make accurate predictions on new, unseen data.

Key Steps in Model Training

a. Prepare the Data

- **Collect data**: Use labeled data for supervised learning.
- **Preprocess**:
- Handle missing values
- Normalize/scale numerical features
- Encode categorical data (e.g., label encoding, one-hot)
- Split into:
    - **Training set** (e.g., 70–80%)
    - **Test set** (e.g., 20–30%

b. Choose an Algorithm

- Based on the type of problem:
  - Classification → Logistic Regression, Random Forest, etc.
  - Regression → Linear Regression, SVR, etc.

c. Train the Model

- Use training data to let the model **learn relationships** between features and labels.

d. Evaluate the Model

- Use the **test set** to evaluate how well the model performs on new data.
- Use appropriate **metrics**:
  - Classification: Accuracy, Precision, Recall, F1-score
  - Regression: MSE, RMSE, R^2 Score

KEY CONCEPTS

1. Bias

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict.

A model with high bias pays very little attention to the training data and oversimplifies the model.

Simple definition: "Resulted Error from Training Data!"

2. Variance

Variance is the variability of a model prediction for a given data point or a value that indicates the spread of our data.

A model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before.

Simple definition: "Resulted Error from Test Data!"

OVERFITTING

Definition: Overfitting occurs when a statistical model or machine learning algorithm learns the noise in the training data rather than the actual pattern.

Symptoms:

- Very low training error but high testing error.
- Low bias, but high variance.

Causes:

- Excessively complex model (too many parameters/features).
- Too little training data.

Consequences:

- Poor generalization to unseen/test data.

Prevention Techniques:

- Use simpler models.
- Cross-validation to evaluate model performance.
- Regularization techniques (e.g., L1/L2 penalties).
- Pruning in decision trees.
- Collect more training data

UNDERFITTING

Definition: Underfitting occurs when a model fails to capture the underlying trend of the data.

Symptoms:

- High error on both training and testing data.
- High bias, but low variance.
- Model is too simple (insufficient complexity).
- Too few features or poor feature selection.
- Inadequate training time or insufficient learning.

Consequences:

- Model performs poorly on all data.
- Fails to learn meaningful patterns.

Prevention Techniques:

- Use a more complex model.
- Add more features or improve feature engineering.
- Reduce regularization if it's too strong.
- Train the model longer or tune hyperparameters.

Both overfitting and underfitting lead to poor predictions on new data sets.

BIAS AND VARIANCE IN THE CASE OF **CLASSIFICATION** MODELS

Model 1:

Training Error = 2%

Testing Error = 18%

⇒ "Low Bais, High Variance".

★ Over-Fitting ★

Model 2:

Training Error = 24%

Testing Error = 28%

⇒ "High Bais, High Variance"

★ Under-Fitting ★

Model 3:

Training Error
= <10%

Testing Error
= <10%

$\Rightarrow$

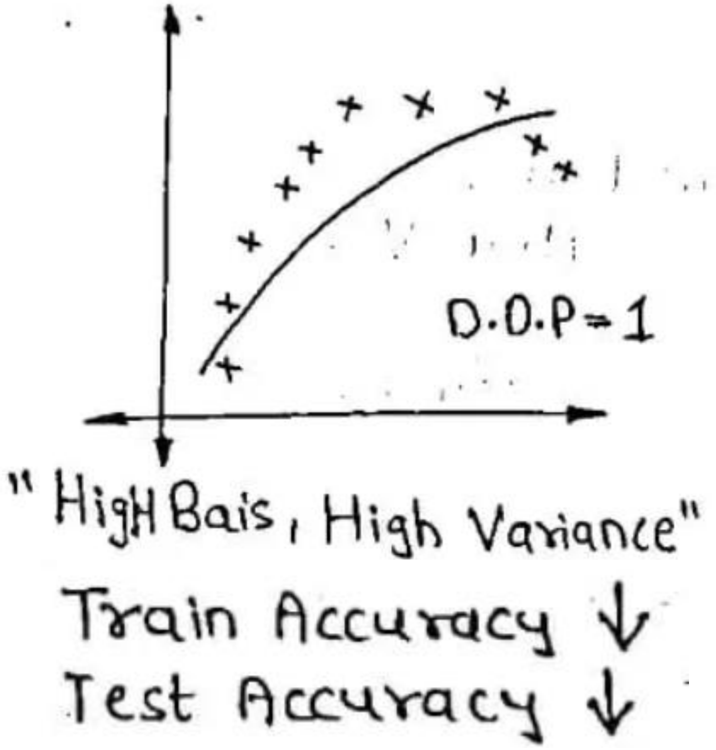
"Low Bais,
Low Variance"

*Most Genralized
Model*

Recommended!

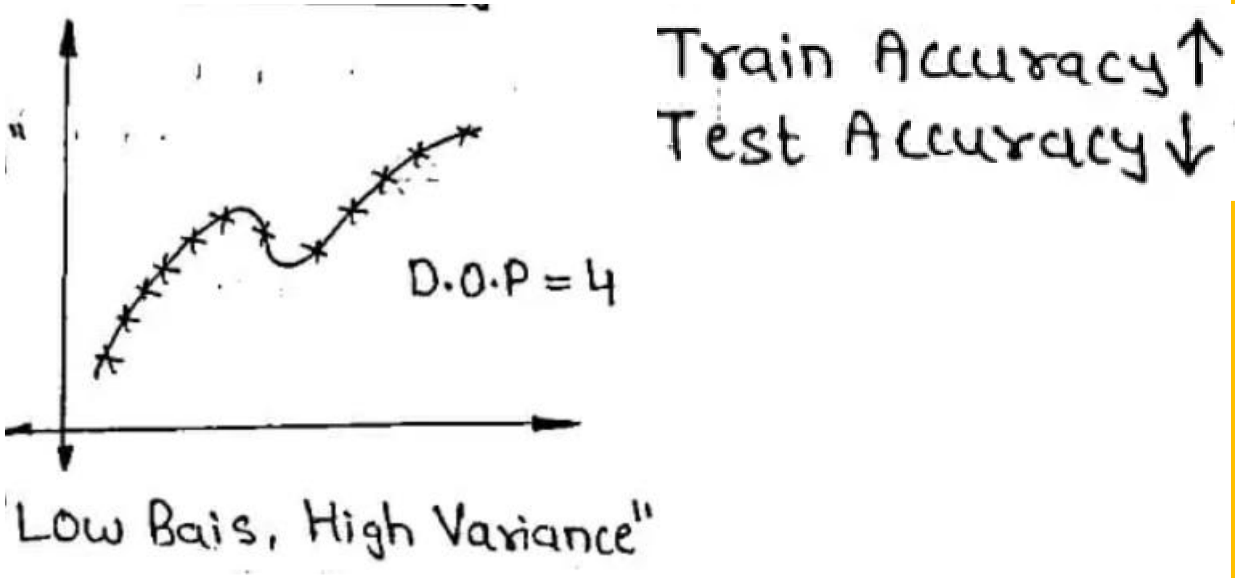
BIAS AND VARIANCE IN THE CASE OF **REGRESSION** MODELS

Model 1: Underfitting



Model-1 will have less train and test accuracy, I.e. Will have High Bias(High Training error) and High Variance(High Testing error).

Model 2: Overfitting

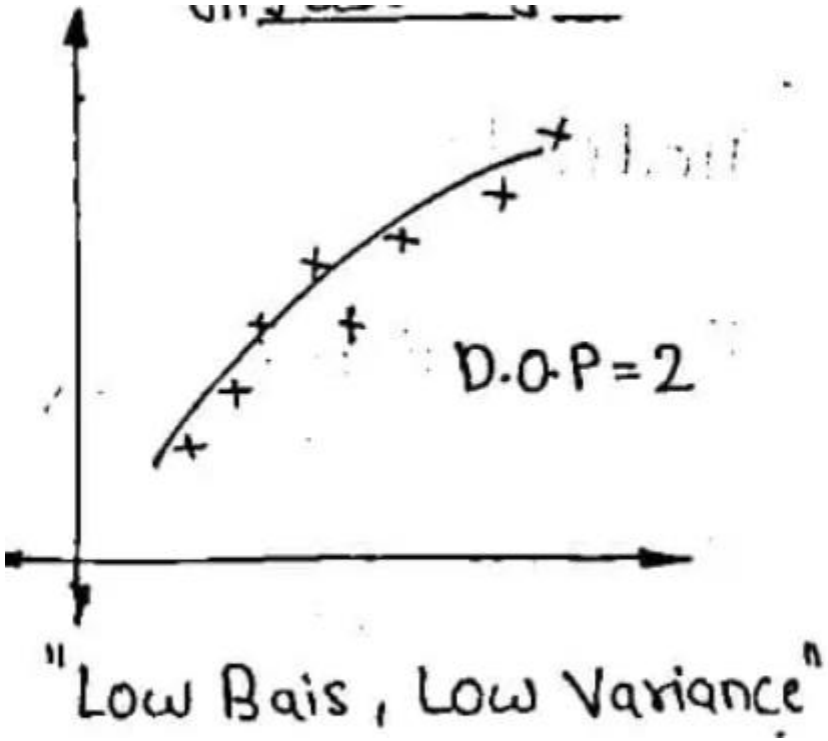


The model2 has trained too good on training data, the reason it fails for testing data(**Low test accuracy**). Since the training accuracy for Model-2 is High and Test accuracy is low, Model-2 will have Low Bias(**Low Training error**) and High Variance(**High Testing error**).

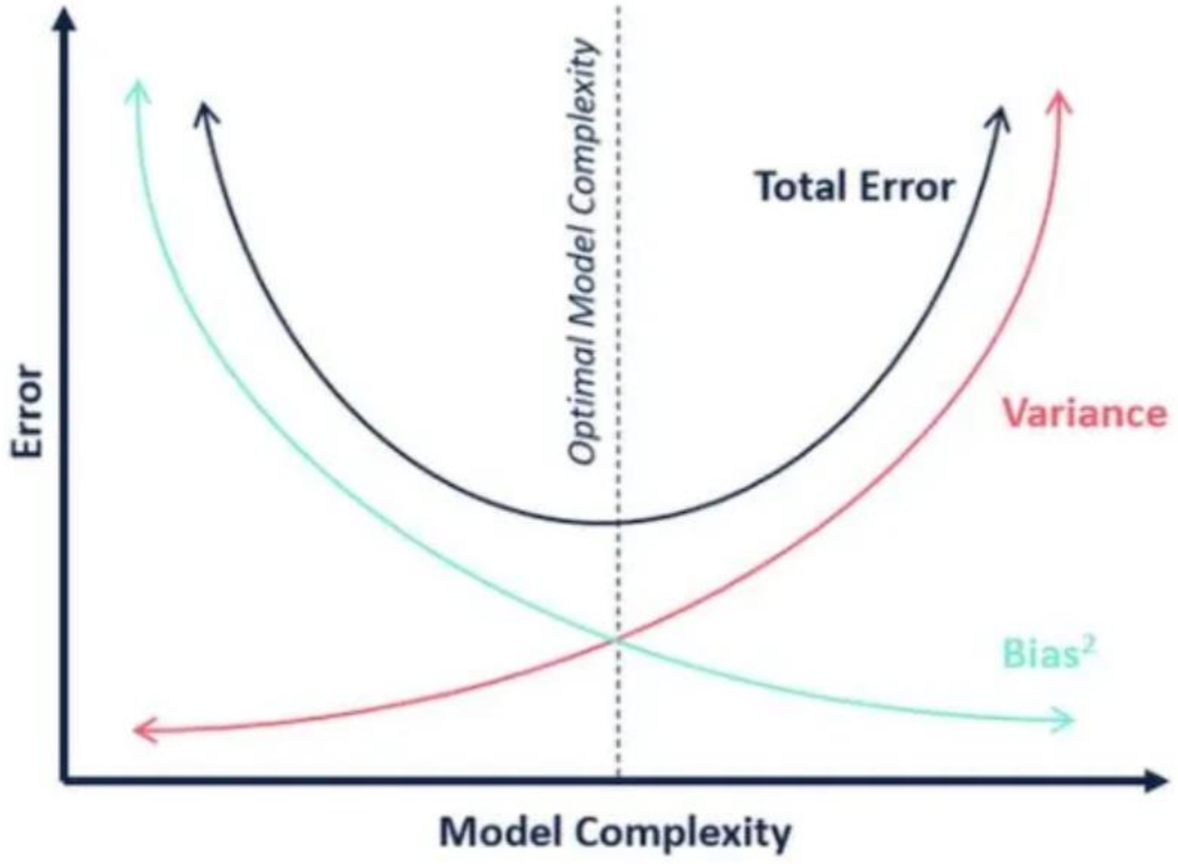
Model 3: Recommended

The model has trained well on training as well as a test set respectively. The reason, model has High training accuracy.

(Low Bias-low training error) and High testing accuracy(Low Variance-low testing error).



TRADE OFF BETWEEN BIAS AND VARIANCE



MODEL EVALUATION

To measure how well a machine learning model performs on **new, unseen data** — not just the data it was trained on.

This ensures that the model is **generalizing** well and not just memorizing.

Why is Evaluation on Unseen Data Important?

- Detects **overfitting** (model performs well on training but poorly on new data)
- Helps **select the best model** among many
- Gives an estimate of **real-world performance**

EVALUATION METRICS FOR CLASSIFICATION

1. Confusion matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Term	**Full Form**	**Meaning**
TP	True Positive	Model predicted **Positive**, and it was **actually Positive**.
TN	True Negative	Model predicted **Negative**, and it was **actually Negative**.
FP	False Positive	Model predicted **Positive**, but it was **actually Negative**. (**Type I error**)
FN	False Negative	Model predicted **Negative**, but it was **actually Positive**. (**Type II error**)

Example: Medical Test for Disease

Let's say a machine learning model is used to detect if a person has a disease.

Person	Actual Condition	Predicted Result	Category
A	Has disease	Has disease	**TP (**Model correctly identifies a sick person.)
B	No disease	No disease	**TN (**Model correctly identifies a healthy person)
C	No disease	Has disease	**FP** (False alarm) (Model wrongly labels a healthy person as sick)
D	Has disease	No disease	**FN** (Model wrongly labels a sick person as healthy (missed detection)

CLASSIFICATION METRICS

Metric	Formula	Description	Range
Accuracy	TP+TN/TP+TN+FP+FN	Proportion of correctly predicted instances.	[0, 1]
Precision	TP/ TP + FP	Correct positive predictions out of all predicted positives.	[0, 1]
Recall (Sensitivity)	TP/TP+FN	Correct positive predictions out of all actual positives.	[0, 1]
F1 Score	$2 \times \frac{Precision \times Recall}{Precision + Recall}$	Harmonic mean of Precision and Recall.	[0, 1]
Specificity	TN/TN+FP	Proportion of correctly predicted negatives.	[0, 1]
ROC-AUC Score	Area under the ROC curve	Measures model's ability to distinguish between classes.	[0, 1]
Confusion Matrix	Tabular form: [[TP, FP], [FN, TN]]	Shows correct and incorrect predictions by class.	
–			

EVALUATION METRICS FOR REGRESSION MODELS

i) Mean Absolute Error (MAE):

- It measures the average absolute difference between the actual and the predicted values.
- As MAE is considered scale-dependent accuracy (sensitive to the scale of the data), it cannot be used to compare the MAE results of the models on datasets with different scales.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y_i}|$$

ii) Mean Square Error (MSE):

It measures the average square of the differences between the predicted scores and the actual scores.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \hat{y_i}\right)^2$$

EVALUATION METRICS FOR REGRESSION MODELS

iii) Root Mean Square Error (RMSE):

- The square root of MSE.
- A popular regression metric used in literature that helps to compare and contrast the results across studies and models.
- It is sensitive to the outliers and influenced by the target variable scale.

$$RMSE = \sqrt{MSE}$$

iv) Normalized Root Mean Square Error (NRMSE):

- The **RMSE** is an absolute error metric **closely related to the absolute scale of data distribution.** This nature of RMSE makes it difficult to directly compare and assess the model's performance on various datasets with different scales.
- The NRMSE scales the RMSE by the range of the data and provides a normalized measure facilitating comparison across the datasets with diverse scales.

$$NRMSE = \frac{RMSE}{Range\ of\ data}$$

METRICS FOR CLUSTERING

1. Silhouette Score

The **Silhouette Score** is a way to measure how good the clusters are in a dataset.

It helps us understand how well the data points have been grouped. The score ranges from -1 to 1.

Silhouette Score (S) for a data point i is calculated as:

$$S(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

- a(i) is the average distance from i to other data points in the same cluster.
- b(i) is the smallest average distance from i to data points in a different cluster.

2. Davies-Bouldin Index

The **Davies-Bouldin Index (DBI)** helps us measure how good the clustering is in a dataset. It looks at how tight each cluster is (compactness), and how far apart the clusters are (separation).

- Lower DBI = better, clearer clusters
- Higher DBI = messy, overlapping clusters

Davies-Bouldin Index (DB) is calculated as:

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\left(\frac{R_{ii}+R_{jj}}{R_{ij}}\right)$$

where,

- k is the total number of clusters.
- R_{ii} is the compactness of cluster i.
- R_{jj} is the compactness of cluster j.
- R_{ij} is the dissimilarity (distance) between cluster i and cluster j.