19UCSPEX02
Machine Learning Techniques

INTRODUCTION

OBJECTIVES:

- Explore the basic concepts of Machine Learning.
- Introduce supervised learning algorithms to build models.
- Gain knowledge about ensemble and unsupervised learning algorithms.
- Learn the basics of deep learning using neural networks.
- Acquire knowledge on graphical models.

Course Outcomes

- Explain the basic concepts of machine learning.
- Design supervised learning models to predict outcomes.
- Develop ensemble and unsupervised models.
- Build deep learning neural network models.
- Apply graphical models to represent complex domains using probability.

Syllabus

- UNIT I - INTRODUCTION TO MACHINE LEARNING
- UNIT II - SUPERVISED LEARNING
- UNIT III - UNSUPERVISED LEARNING
- UNIT IV - NEURAL NETWORKS
- UNIT V - PROBABILISTIC GRAPHICAL MODELS

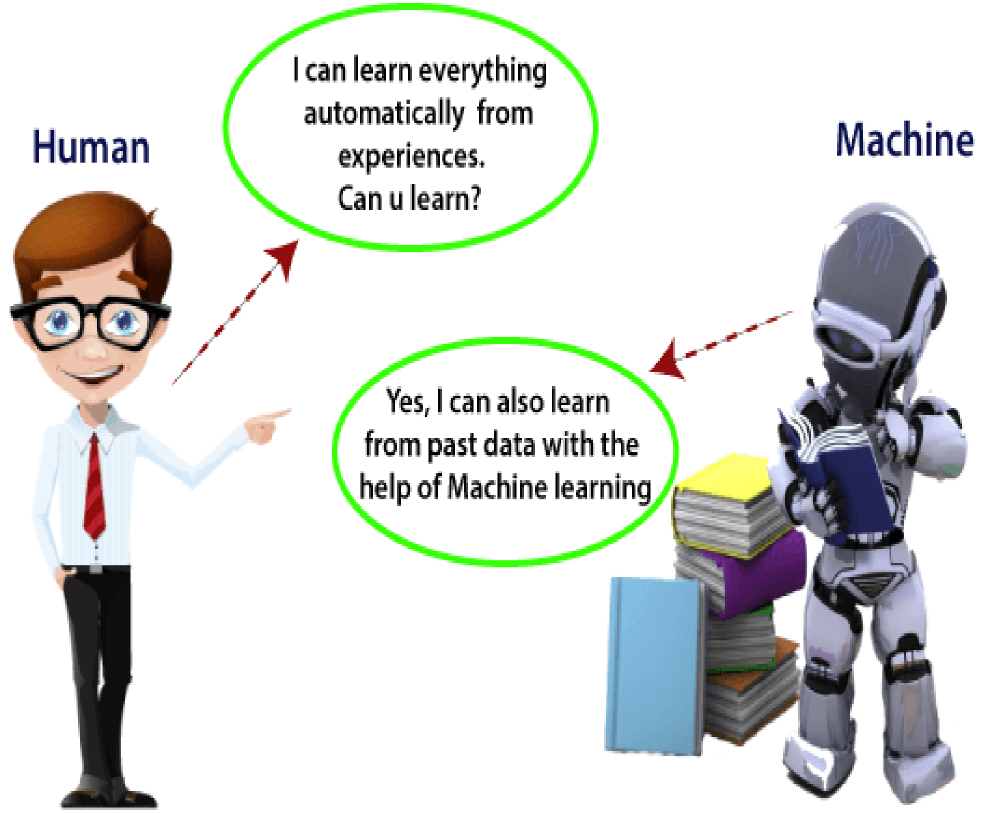
Text Books

1. Ethem Alpaydin, “Introduction to Machine Learning”, MIT Press, Fourth Edition, 2020.

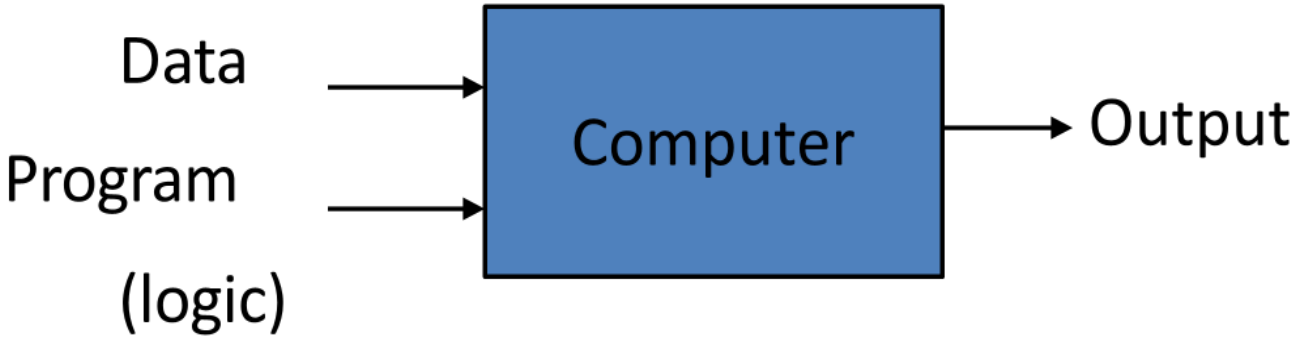
2. Stephen Marsland, “Machine Learning: An Algorithmic Perspective, “Second Edition”, CRC Press, 2014.

Machine Learning?

- Machine learning is a type of artificial intelligence (AI) that allows computers to learn and improve from experience without being explicitly programmed.
- It uses algorithms to analyze large amounts of data, identify patterns, and make predictions


Human
I can learn everything automatically from experiences. Can u learn?
Machine
Yes, I can also learn from past data with the help of Machine learning

Traditional Programming



Data

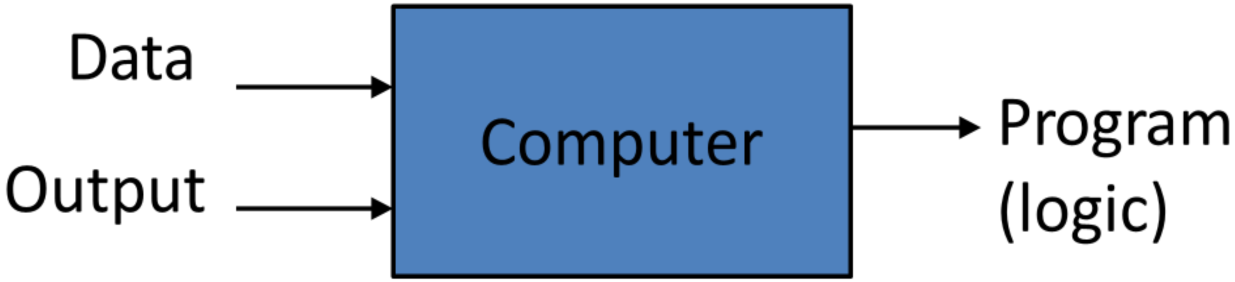
Program

(logic)

Computer

Output

Machine Learning



Data

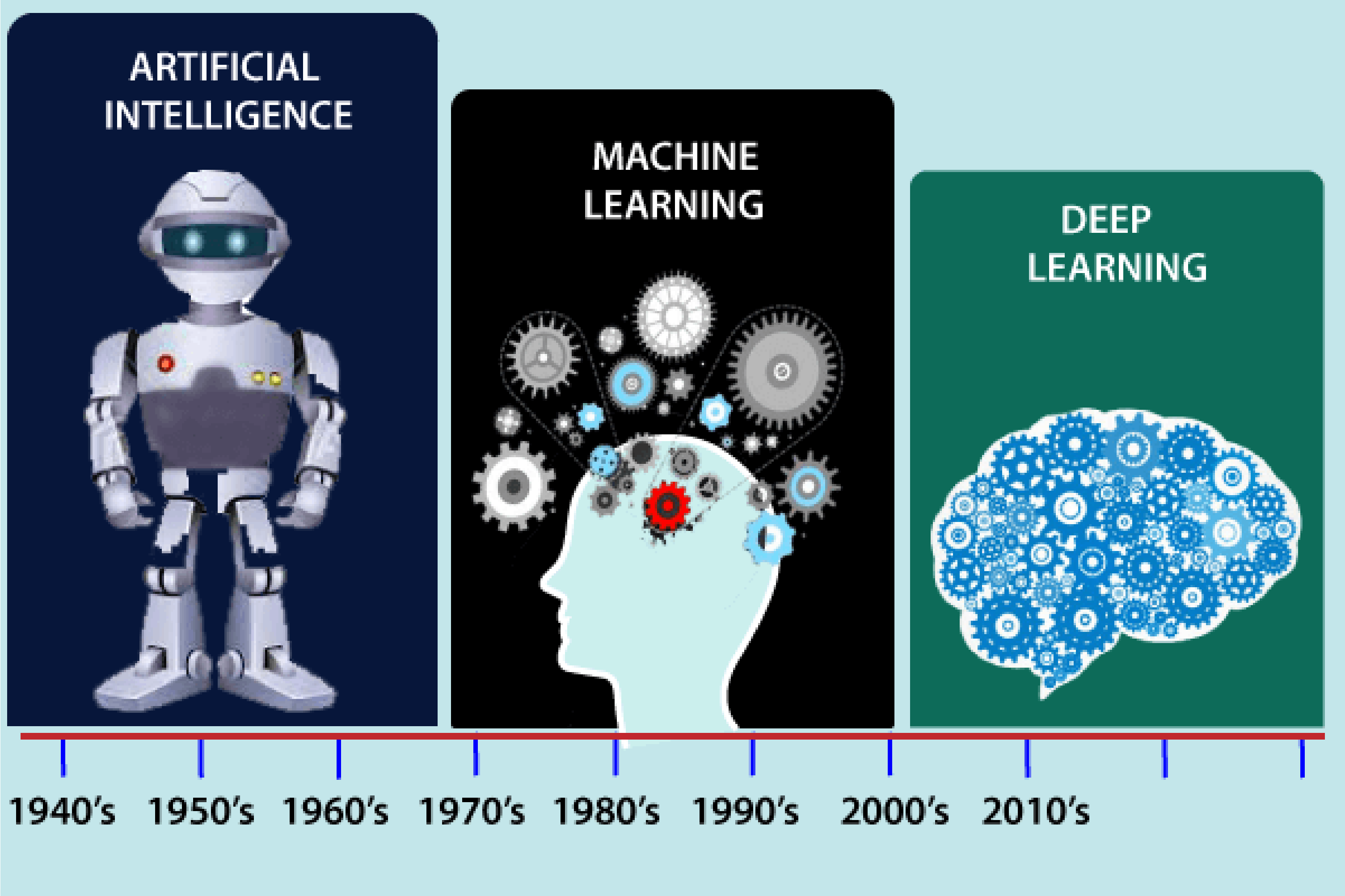
Output

Computer

Program

(logic)

History of Machine Learning



Features of Machine Learning:

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

Examples of machine learning

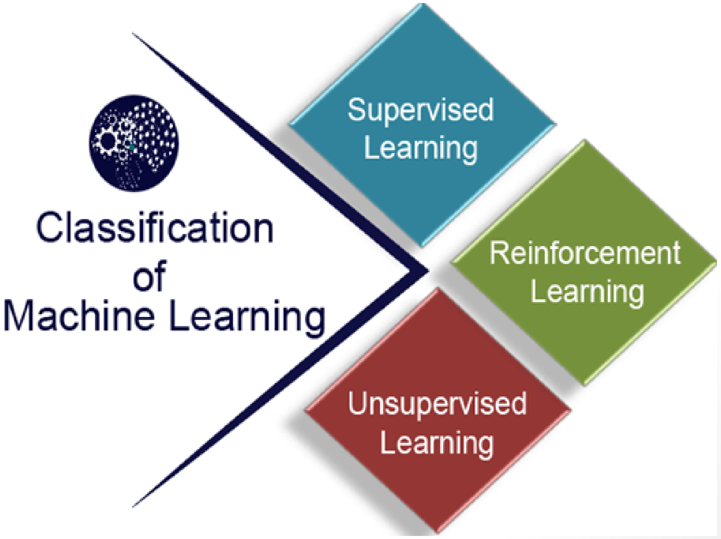
- **Speech recognition**: Uses natural language processing (NLP) to translate human speech into text
- **Computer vision**: Uses AI to analyze images and videos to derive information
- **Recommendation engines**: Uses data on past consumption to develop cross-selling strategies
- **Fraud detection**: Uses machine learning to identify suspicious transactions
- **Natural language processing**: Uses machine learning to understand natural language, create new text, and translate between languages

I- INTRODUCTION TO MACHINE LEARNING

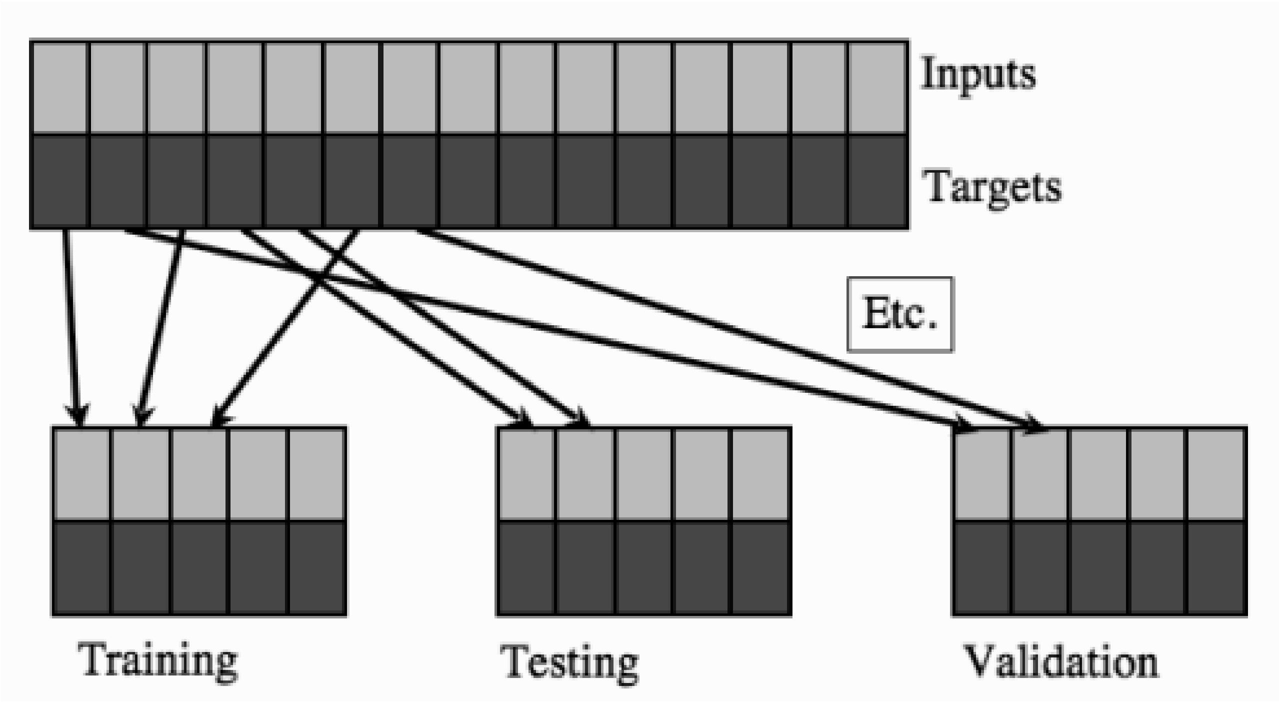
Machine Learning

- Machine Learning (ML) is a data analysis technique that **allows an analytic system to learn** through the course of solving many similar problems.
- Machine learning is a subset of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn from and make predictions or decisions based on data

Machine Learning Types:



Testing machine learning algorithms



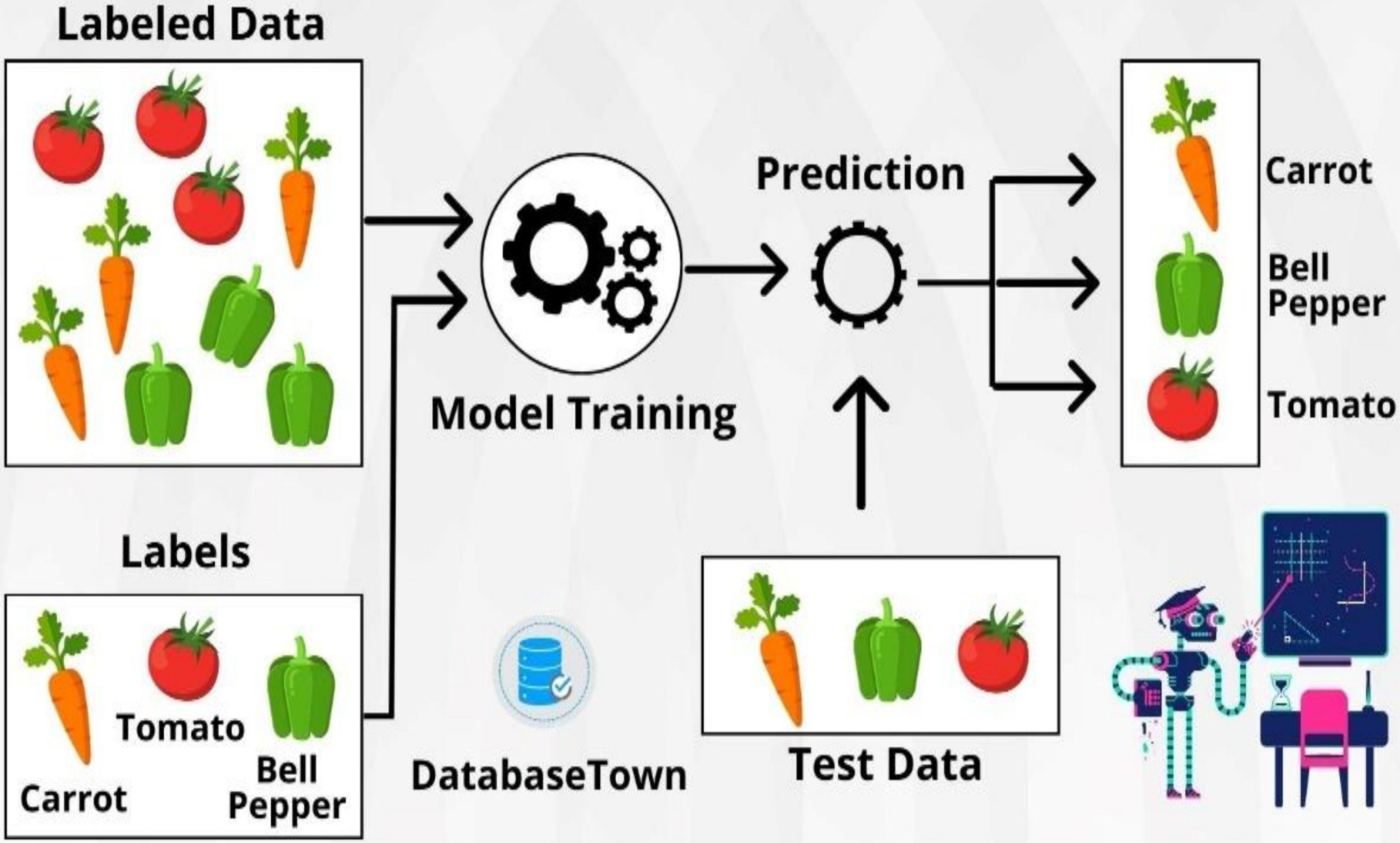
II-SUPERVISED LEARNING

- Supervised learning is a type of machine learning algorithm that learns from labeled data.

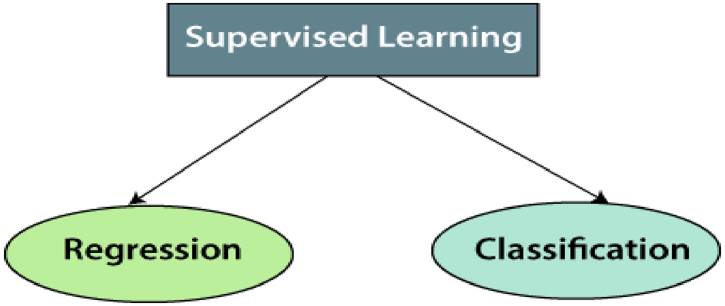
- Labeled data is data that has been tagged with a correct answer or classification

SUPERVISED LEARNING

Supervised machine learning is a branch of artificial intelligence that focuses on training models to make predictions or decisions based on labeled training data.

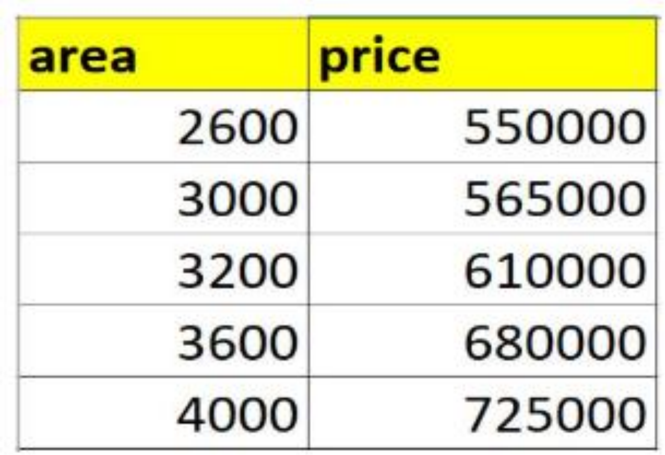


Regression

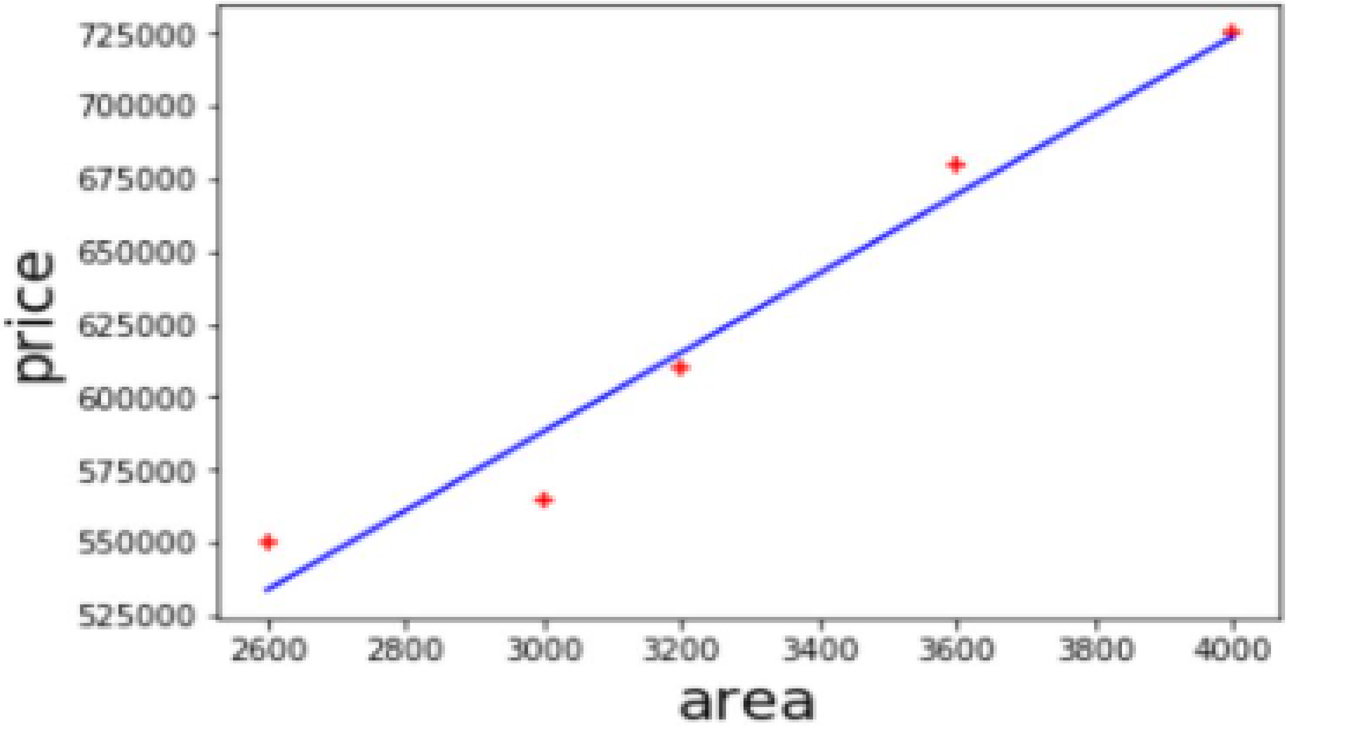


- **Regression** is refers to a supervised learning technique where the goal is to predict a continuous numerical value based on one or more independent features.
- **Types of Regression**
- Linear Regression.
- Polynomial Regression.
- Logistic Regression.
- Multiple Linear Regression

Example for Linear Regression



area	price
2600	550000
3000	565000
3200	610000
3600	680000
4000	725000



Classification

- The Classification algorithm is a Supervised Learning technique that is **used to identify the category of new observations** on the basis of training data.

Types of ML Classification Algorithms:

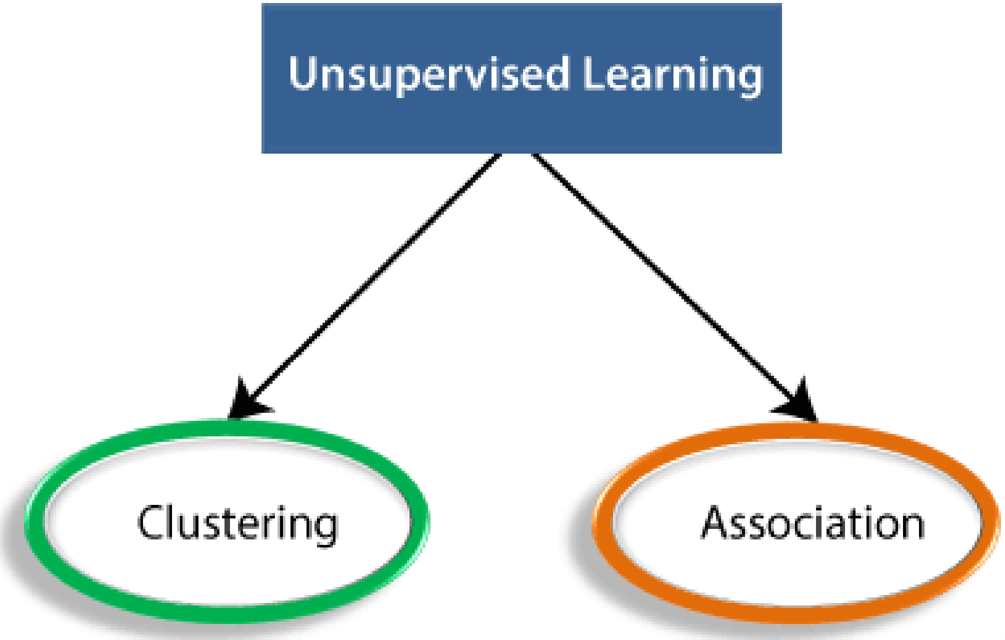
- Naive Bayes
- K-Nearest Neighbours
- Support Vector Machines
- Kernel SVM
- Decision Tree Classification
- Random Forest Classification

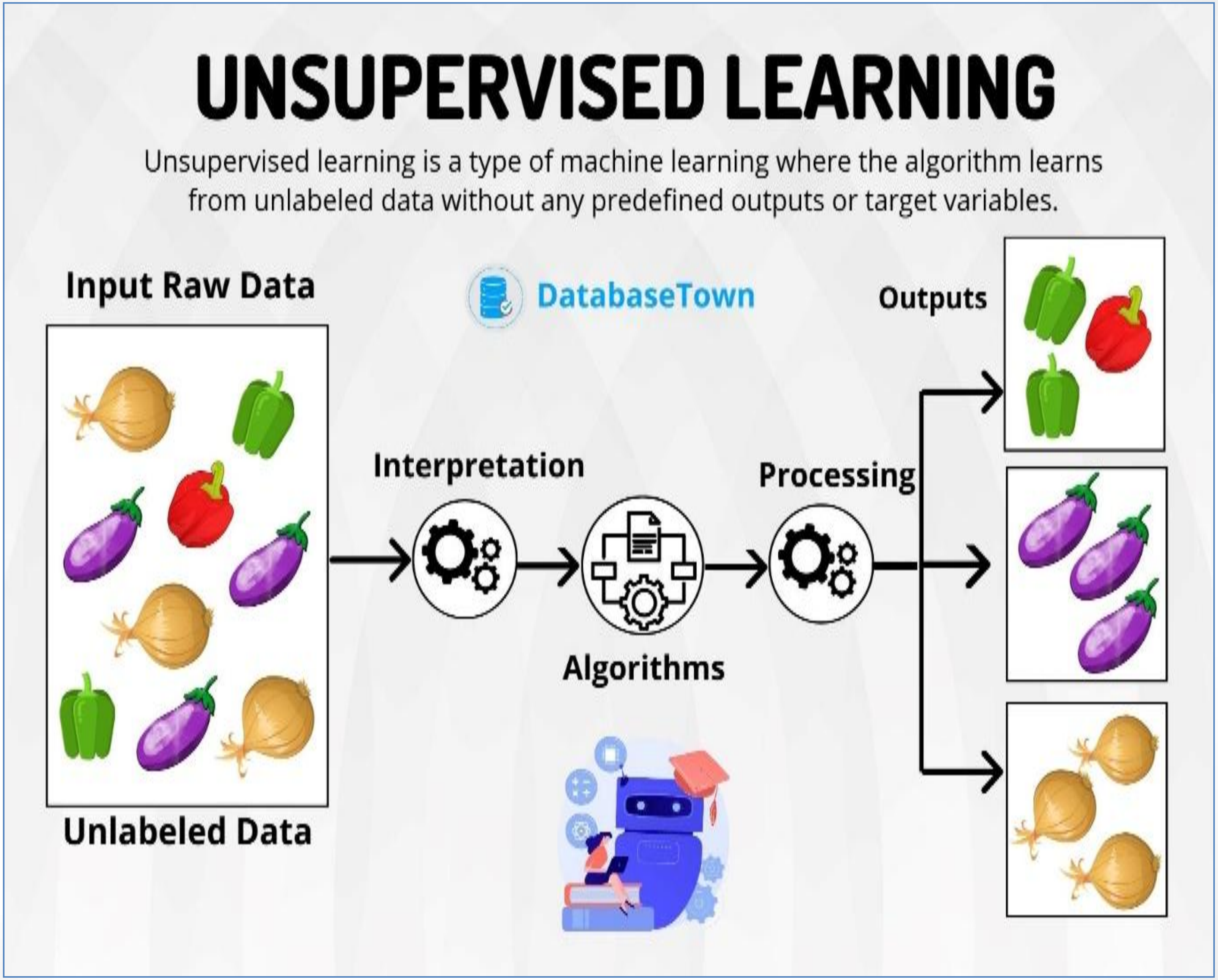
Ensemble Learning

- The idea behind ensemble learning is that by **combining multiple models**, each with its strengths and weaknesses
- The ensemble can achieve better results than any single model alone.
- Ensemble learning can be applied to various machine learning tasks, including **classification, regression, and clustering**.

III UNSUPERVISED LEARNING

- **Unsupervised learning** is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.



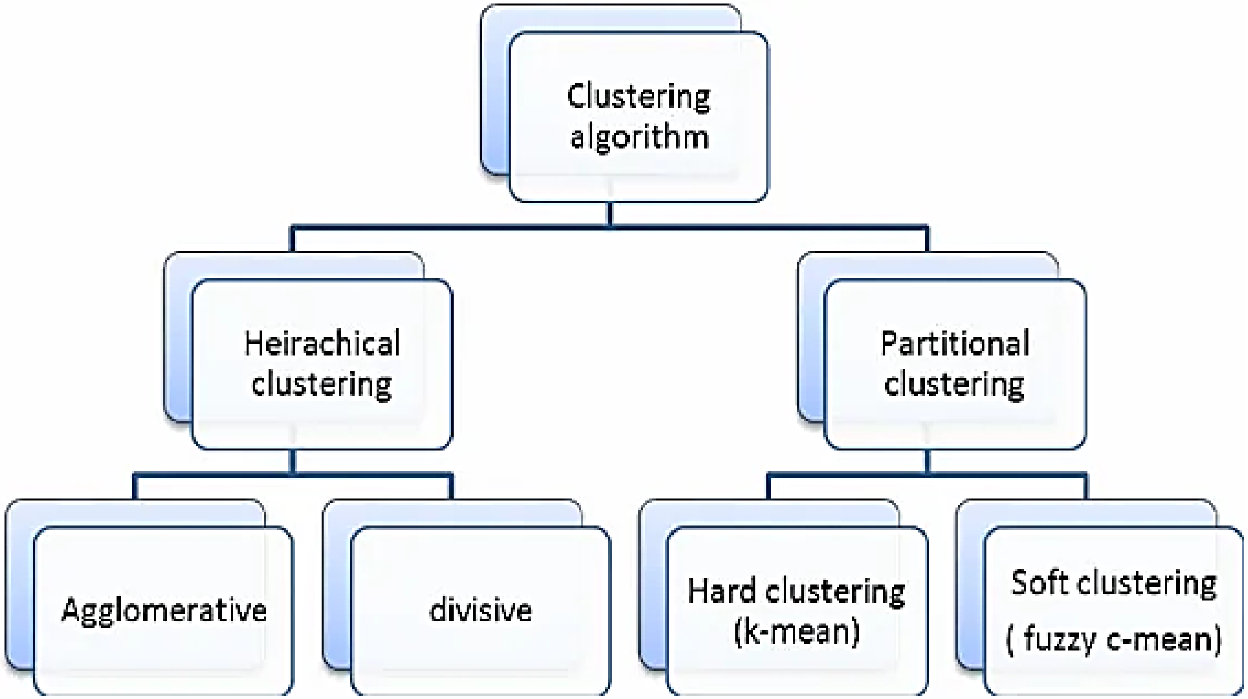


UNSUPERVISED LEARNING

Unsupervised learning is a type of machine learning where the algorithm learns from unlabeled data without any predefined outputs or target variables.

Hierarchical Clustering

Clustering is **an unsupervised machine learning technique** designed to group unlabeled examples based on their similarity to each other.



Dimensionality Reduction:

- Dimensionality reduction is a technique **used to reduce the number of features in a dataset** while retaining as much of the important information as possible.
- Principal components analysis (PCA)
- Locally Linear Embedding (LLE).

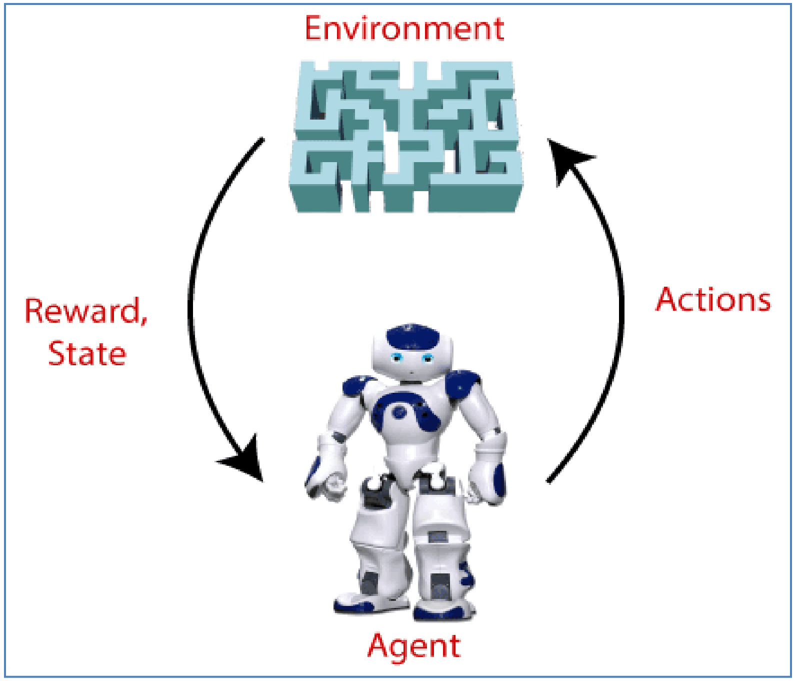
Reinforcement Learning

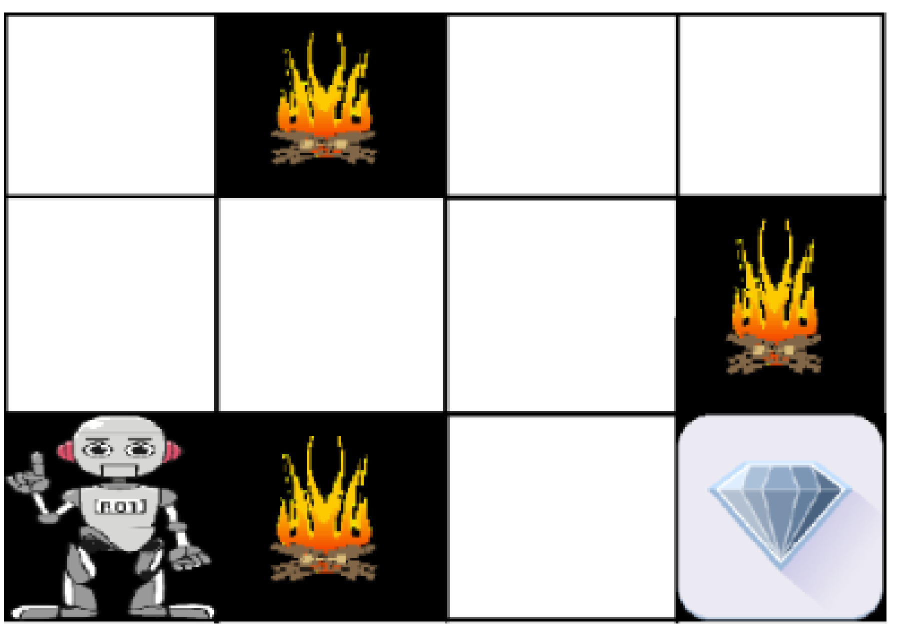
- In Reinforcement Learning (RL) a so-called Agent learns to achieve its goals in an uncertain, potentially complex environment.
- The goal of the agent is to maximize the total reward.

Reinforcement learning

Key Concepts of Reinforcement Learning

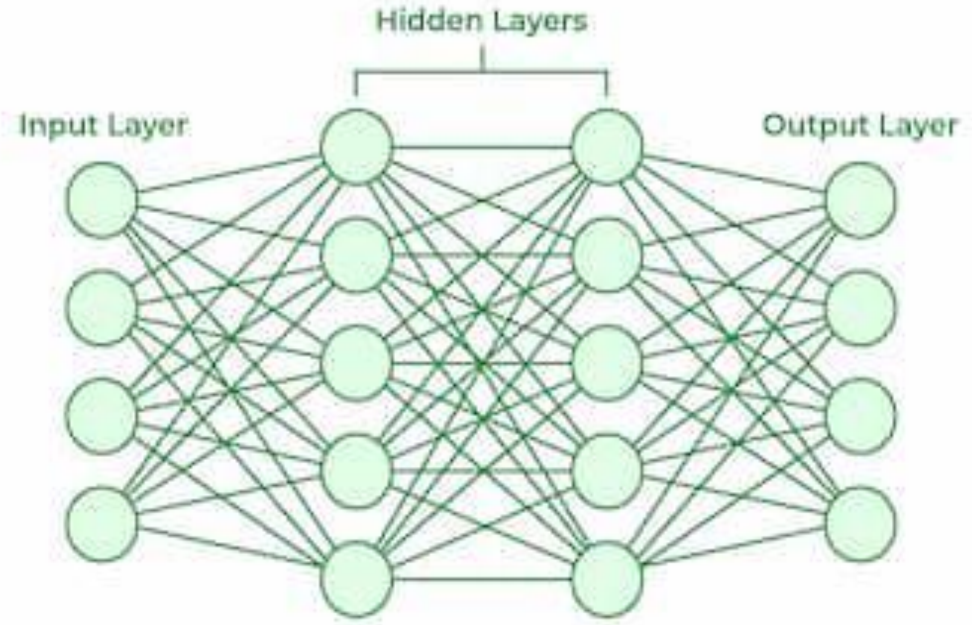
- **Agent:** The learner or decision-maker.
- **Environment:** Everything the agent interacts with.
- **State:** A specific situation in which the agent finds itself.
- **Action:** All possible moves the agent can make.
- **Reward:** Feedback from the environment based on the action taken.


Environment
Reward,
State
Actions
Agent


RO1

IV- NEURAL NETWORKS

- A neural network is a method in (AI) that teaches computers to process data in a way that is inspired by the human brain.
- **Deep learning** is a subset of **machine learning** that uses multilayered neural networks, to simulate the complex decision-making power of the human brain.



Types of Neural Networks

- **Feedforward Networks**: A feedforward neural network is a simple artificial neural network architecture in which data moves from input to output in a single direction.
- **Multilayer Perceptron (MLP)**: It is a type of feedforward network with three or more layers, including an input layer, one or more hidden layers, and an output layer.
- **Convolutional Neural Network (CNN)**: A CNN is a specialized artificial neural network designed for image processing.
- **Recurrent Neural Network (RNN)**: used for sequential data processing is called a Recurrent Neural Network (RNN).
- **Long Short-Term Memory (LSTM)**: It is a type of RNN that is designed to overcome the vanishing gradient problem in training RNNs.

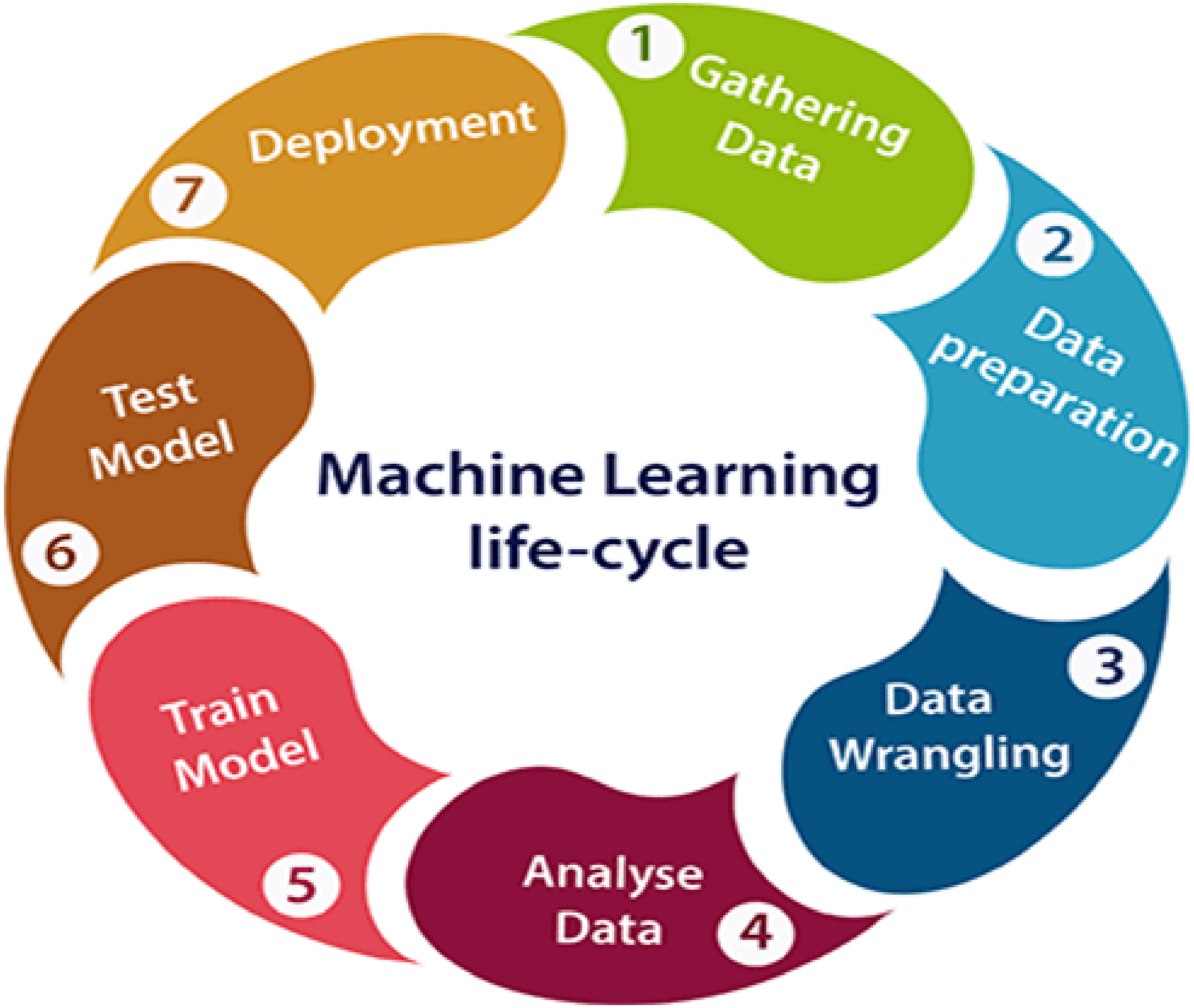
V- PROBABILISTIC GRAPHICAL MODELS

- Probabilistic graphical models (PGMs) are a **framework for representing probability distributions over complex domains.**
- They are used in machine learning and other fields to model the relationships between random variables.
- **Bayesian networks**: A branch of PGMs that model problems as directed acyclic graphs (DAGs)
- **Markov random fields**: A branch of PGMs that model problems as Markov random fields

Hidden Markov Model

- ***A Hidden Markov Model (HMM) is a probabilistic model that consists of a sequence of hidden states, each of which generates an observation***
- It is commonly used in machine learning for tasks such as speech recognition, natural language processing, and bioinformatics.

Machine Learning Process



1.Gathering Data

This step includes the below tasks:

- **Identify various data sources**
- **Collect data**
- **Integrate the data obtained from different sources**

By performing the above task, we get a coherent set of data, also called as a **dataset**.

2. Data preparation

This step can be further divided into two processes:

- **Data exploration:**
  It is used to understand the nature of data that we have to work with. We need to understand the **characteristics, format, and quality of data**.
  A better understanding of data leads to an effective outcome. In this, we find Correlations, general trends, and outliers.
- **Data pre-processing:**
  Now the next step is preprocessing of data for its analysis.

3. Data Wrangling

- Data wrangling is the process of **cleaning and converting raw data into a useable format**.
- In real-world applications, collected data may have various issues, including:
  - **Missing Values**
  - **Duplicate data**
  - **Invalid data**
  - **Noise**
- So, we use various **filtering techniques** to clean the data.
- Splitting the cleaned data into two sets - a **training set and a testing set**.
  - The training set is the set your model learns from.
  - A testing set is used to check the accuracy of your model after training.

4. Data Analysis

This step involves:

- **Selection of analytical techniques**
- **Building models**
- **Review the result**

- It starts with the determination of the type of the problems, where we select the machine learning techniques such as **Classification**, **Regression**, **Cluster analysis**, **Association**, etc. then build the model using prepared data, and evaluate the model.

5. Train Model

- To train the model, data is split into three parts- **Training data, validation data, and test data**.
- Around 70%-80% of data goes into the training data set which is used in training the model.
- **Validation data** is also known as development set or dev set and is used to avoid **overfitting or underfitting** situations i.e. enabling hyperparameter tuning.
- **Hyperparameter tuning** is a technique used to combat overfitting and underfitting.
- Validation data is used during **model evaluation**.
- Around 10%-15% of data is used as validation data.

6. Test Model

- In this step, we check for the accuracy of our model by providing a test dataset to it.
- Rest 10%-20% of data goes into the test data set. Test data set is used for **testing after the model preparation.**
- Common metrics for evaluating a model's performance include **accuracy** (for classification problems), **precision** and recall (for binary classification problems), and **mean squared error** (for regression problems).

7. Deployment

- The last step of machine learning life cycle is **deployment**, where we **deploy the model in the real-world system**.
- If the above-prepared model is producing an accurate result as per our requirement with acceptable speed, then we deploy the model in the real system.

- But before deploying the project, we will check whether it is improving its performance using available data or not. The deployment phase is similar to making the final report for a project.

Preliminaries

- **Inputs**: These are the data fed into the algorithm. They are represented as input vectors, which are lists of real numbers. For example, (0.2, 0.45, 0.75, -0.3) is a 4-dimensional input vector.

- **Weights**: Weights are values that determine the strength of connections between nodes in a neural network. These weights are organized into a matrix called **W**.

- **Outputs**: The result produced by the algorithm after processing the inputs. The output is represented as a vector **y**, and its elements are the values produced by the network for each dimension of the output.

Preliminaries

- **Targets**: In supervised learning, the target vector **t** contains the correct answers (i.e., the expected outputs), which the algorithm learns to predict.

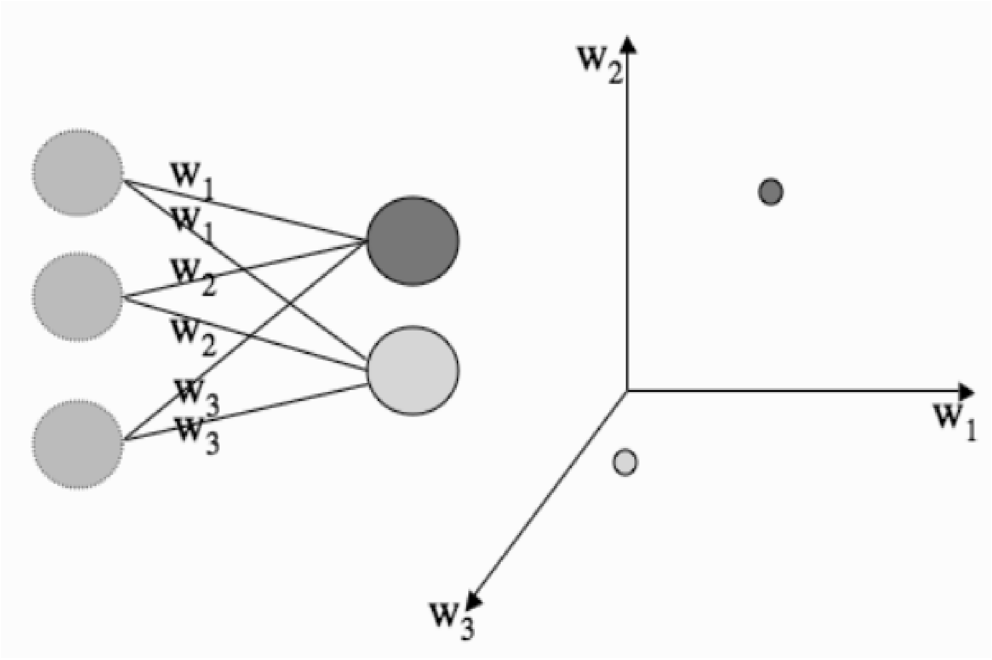
- **Activation Function**: For neural networks, this function **g(·)** defines how the neuron responds to the weighted inputs. It can be a threshold function or another type of function.

- **Error**: The error **E** measures the difference between the algorithm's outputs **y** and the targets **t**, helping to quantify how accurately the network is performing

WEIGHT SPACE

- If our data has only two or three input dimensions, then this is pretty easy: we use
  - 1,the x-axis for feature
  - 2, the y-axis for feature
  - 3, and the z-axis for feature

We then plot the positions of the input vectors on these axes.



THE CURSE OF DIMENSIONALITY

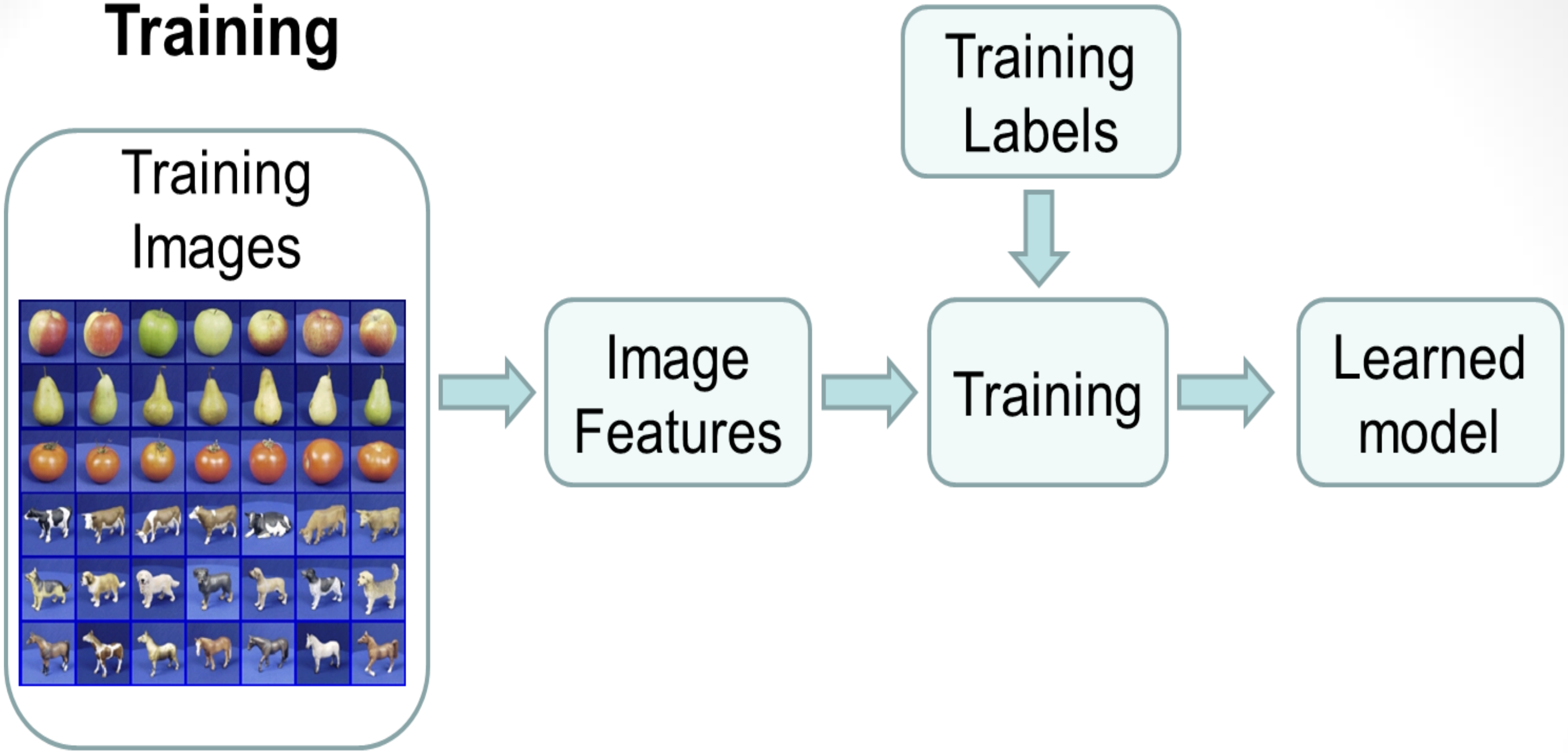
- Curse of Dimensionality refers to the phenomenon where the efficiency and effectiveness of algorithms deteriorate as the dimensionality of the data increases exponentially.
- Curse of Dimensionality significantly impacts machine learning algorithms in various ways. It leads to increased computational complexity, longer training times, and higher resource requirements. Moreover, it escalates the risk of overfitting and spurious correlations.

- Techniques like **dimensionality reduction, feature selection,** and careful model design are essential for mitigating its effects and improving algorithm performance

Testing Machine Learning algorithms

- The purpose of learning is to get better at predicting the outputs, be they class labels or continuous regression values.
- The only real way to know how successfully the algorithm has learnt is to compare the predictions with known target labels, which is how the training is done for supervised learning.

Training



Training Images

Image Features

Training Labels

Training

Learned model

Testing

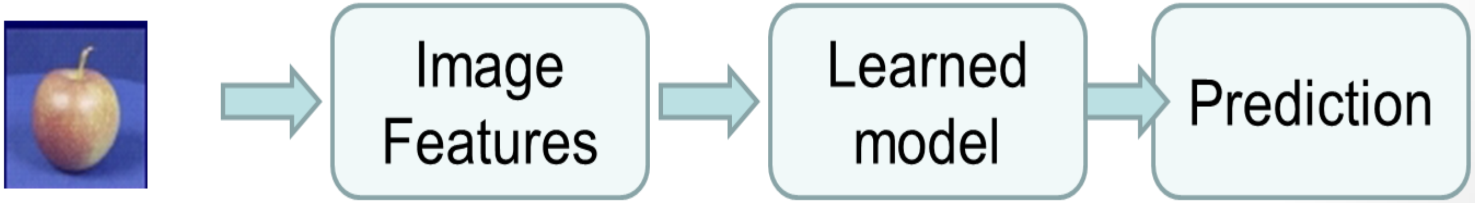
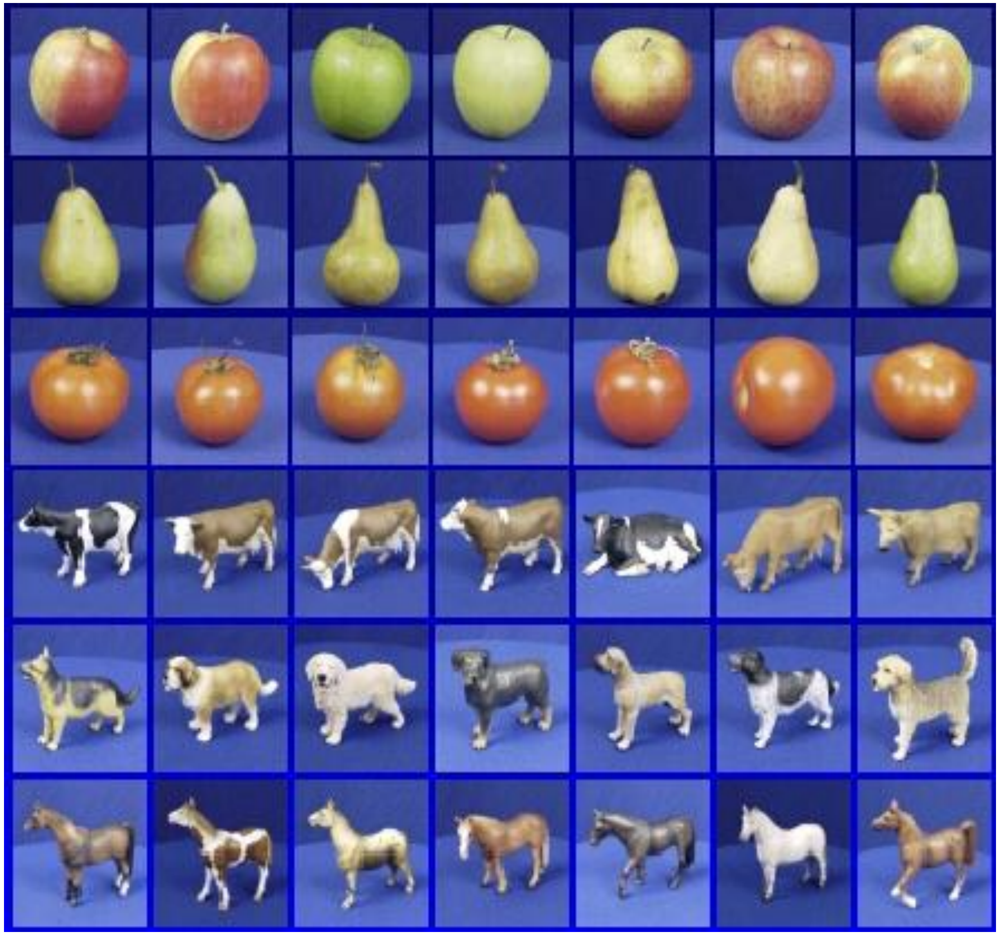


Image Features

Learned model

Prediction

Training and testing data



Training, Testing, and Validation Set

Three sets of data:

- The **training set** to actually train the algorithm,
- The **validation set** to keep track of how well it is doing as it learns, and
- The **test set to** produce the final results.
- This is becoming expensive in data, especially since for supervised learning it all has to have target values attached (and even for unsupervised learning,

Data Splitting

- How you split your data into these three sets can affect the model's performance. A common split ratio is **50:25:25** (50% training, 25% validation, 25% test) if you have a lot of data.For smaller datasets, you might use **60:20:20** (60% training, 20% validation, 20% test).

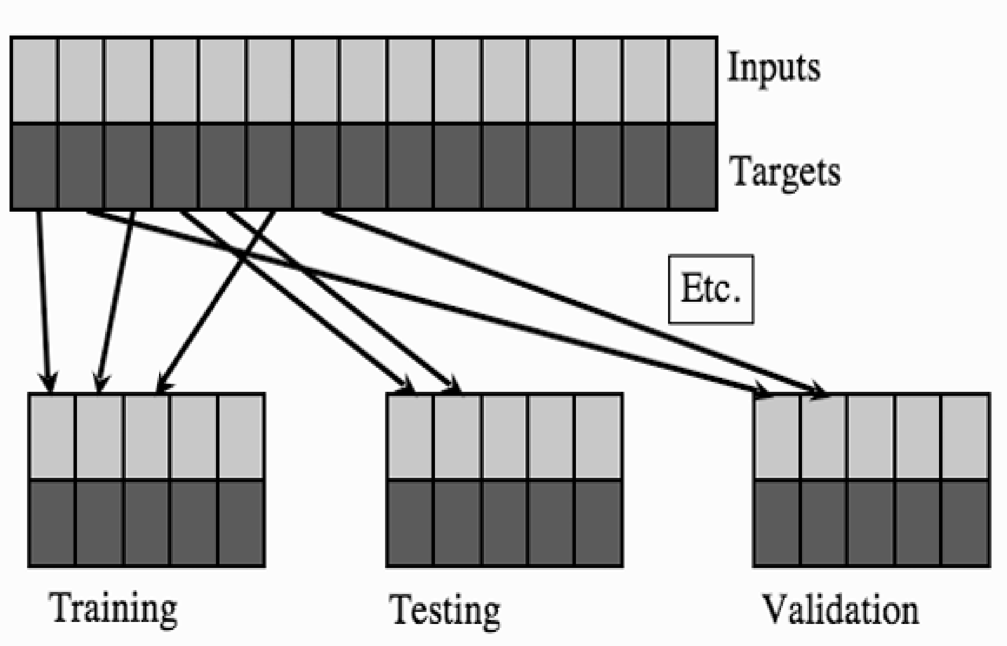
Random Shuffling

- If your dataset has data points in a specific order (e.g., class 1 followed by class 2), it can lead to biased results if you simply split the data sequentially. To avoid this, you can **randomly shuffle** the data before splitting, ensuring the model sees a mix of all classes.

Cross-Validation

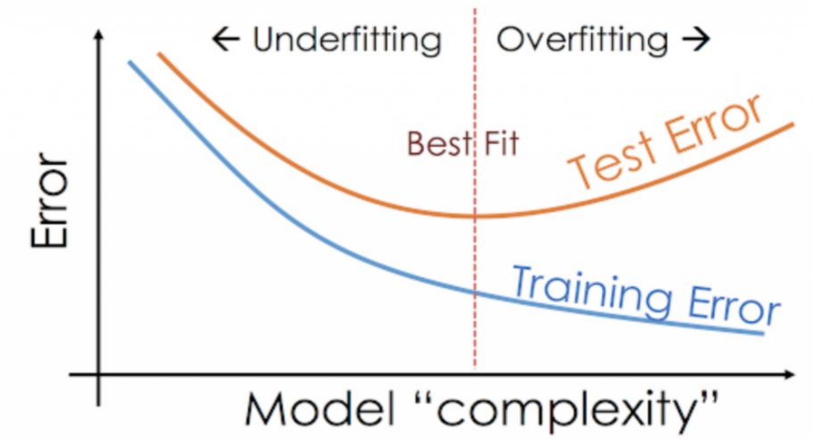
Example of K-fold Cross-Validation: Figure 9

- Split data into 5 subsets (K=5).
- Train the model on 4 subsets, and test on the remaining one.
- Rotate the subsets and repeat the process.



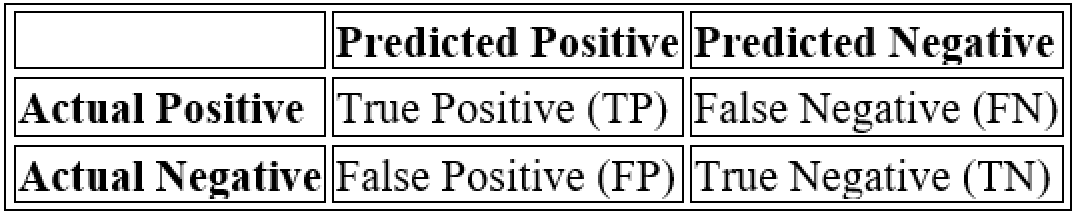
Overfitting

- Overfitting happens when a machine learning model becomes overly intricate, essentially memorizing the training data. While this might lead to high accuracy on the training set, the model may struggle with new, unseen data due to its excessive focus on specific details.



Confusion Matrix

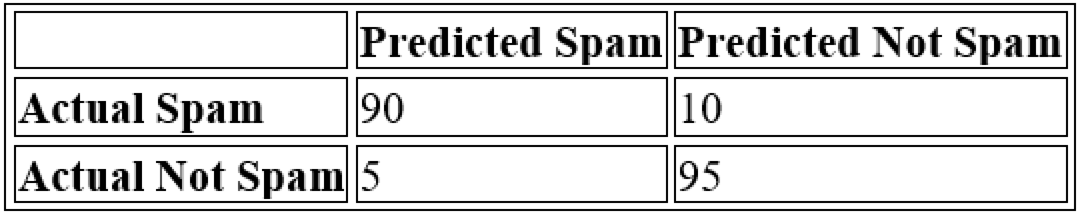
- A **confusion matrix** is a table used to evaluate the performance of a classification algorithm. It compares the predicted labels with the true labels (actual labels) from the data



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

Example of a Confusion Matrix:

- Imagine a model classifying whether an email is **spam** or **not spam**:



	Predicted Spam	Predicted Not Spam
Actual Spam	90	10
Actual Not Spam	5	95

- **True Positives (TP)** = 90 (correctly classified spam emails)
- **False Negatives (FN)** = 10 (spam emails incorrectly classified as not spam)
- **False Positives (FP)** = 5 (non-spam emails incorrectly classified as spam)
- **True Negatives (TN)** = 95 (correctly classified non-spam emails)

Accuracy

- Accuracy is most common performance metric for classification algorithms. It may be defined as the number of correct predictions made as a ratio of all predictions made.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions}$$

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

$$\text{Accuracy} = \frac{90 + 95}{90 + 95 + 5 + 10} = \frac{185}{200} = 0.925$$

Precision

- Precision measures the proportion of true positive instances out of all predicted positive instances. It is calculated as the number of true positive instances divided by the sum of true positive and false positive instances.

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

Recall or Sensitivity

- Recall measures the proportion of true positive instances out of all actual positive instances. It is calculated as the number of true positive instances divided by the sum of true positive and false negative instances.
- **True Positive Rate** is considered as a portion of positive data points that are correctly considered as positive, with respect to all data points that are positive.
- **Measures how well the model identifies positive instances.**

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

Specificity

- Specificity, in contrast to recall, may be defined as the number of negatives returned by our ML model.
- **False Negative Rate** is considered as a portion of negative data points that are correctly considered as negative, with respect to all data points that are negatives.

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

Specificity

- Specificity, in contrast to recall, may be defined as the number of negatives returned by our ML model.
- **False Negative Rate** is considered as a portion of negative data points that are correctly considered as negative, with respect to all data points that are negatives.

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

F1 Score

- F1 score is the harmonic mean of precision and recall. It is a balanced measure that takes into account both precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 would be 1 and worst would be 0.
- It is used to evaluate the performance of a classification model, especially in situations where the class distribution is imbalanced

$$F1 - score = 2 * \frac{precision * recall}{precision + recall}$$

ROC Curves

- The **Receiver Operating Characteristic (ROC) Curve** is a graphical representation used to evaluate the performance of a **binary classification** model, particularly in cases where the data may be imbalanced.

- It illustrates the trade-off between **True Positive Rate (TPR)** and **False Positive Rate (FPR)** across different classification thresholds.

False Positive Rate (FPR):

- Measures how often the model incorrectly labels a negative instance as positive.
- FPR is calculated as:

False Positive Rate (FPR):

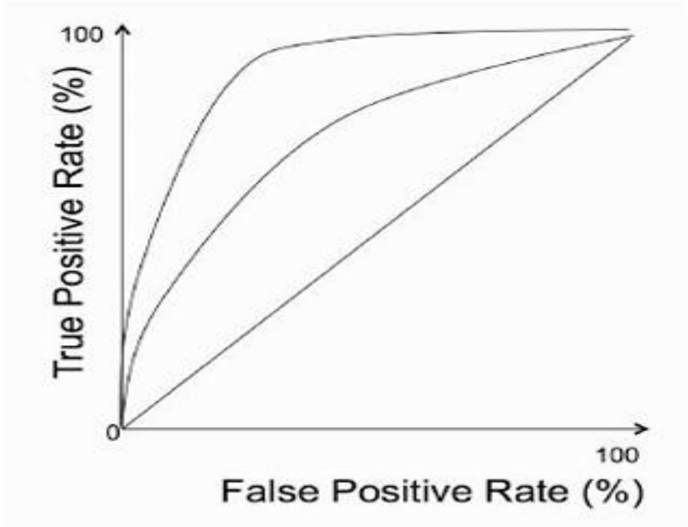
- Measures how often the model incorrectly labels a negative instance as positive.
- FPR is calculated as:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

ROC Curve

Interpreting the ROC Curve

- **Diagonal Line (Random Model)**: A classifier that randomly guesses has an ROC curve that is a diagonal line from (0, 0) to (1, 1). This represents a model that is no better than random chance. A good classifier should be above this line.



ROC Curve

- **Closer to the Top Left Corner**: The closer the ROC curve is to the top-left corner (where TPR = 1 and FPR = 0), the better the model's performance. This indicates high recall (correctly identifying positive cases) and low false positive rate (incorrectly classifying negative cases as positive).
- **Area Under the Curve (AUC)**: The area under the ROC curve (AUC) is an important metric. AUC provides a single scalar value that summarizes the performance of the classifier.
  - **AUC = 1**: Perfect classifier (no false positives or false negatives).
  - **AUC = 0.5**: Random classifier (no better than random guessing).
  - **AUC < 0.5**: The model is worse than random guessing (it consistently makes incorrect predictions)

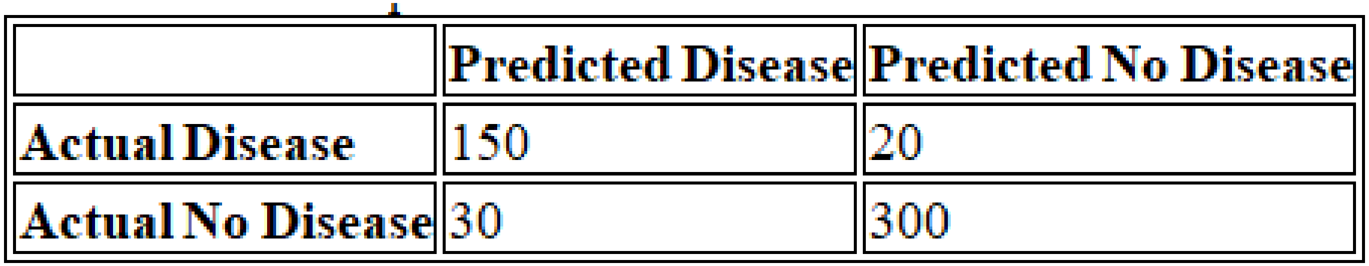
Matthew's Correlation Coefficient (MCC)

- A more robust metric to evaluate model performance on unbalanced datasets is the **Matthew's Correlation Coefficient (MCC)**. It considers all four confusion matrix values (True Positives, True Negatives, False Positives, and False Negatives), providing a balanced measure even when the dataset is highly imbalanced

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Example:

- A machine learning model is used to predict whether a patient has a particular disease (positive) or does not have the disease (negative). The goal is to predict the disease presence based on patient data.
- The confusion matrix for 500 patients is as follows:



	Predicted Disease	Predicted No Disease
Actual Disease	150	20
Actual No Disease	30	300

- **Based on the confusion matrix, calculate the following evaluation metrics for the disease detection model**
- Accuracy
- Precision (Positive Predictive Value)
- Recall (Sensitivity)
- F1 Score
- TPR and FPR under ROC curve
- Calculate MCC
- Interpret the results from the above metrics.

Statistics for Machine Learning

- **Mean (Average):** The sum of all data points divided by the number of points. Used to find the central tendency of the data.

$$\text{Mean} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- **Median:** The middle value in a dataset when arranged in ascending or descending order.
  - More robust than the mean, especially when the data has outliers.
- **Mode:** The most frequently occurring value in a dataset.Useful for categorical data.

Statistics for Machine Learning (Cont..)

- **Variance:** Measures the spread of the data points from the mean.

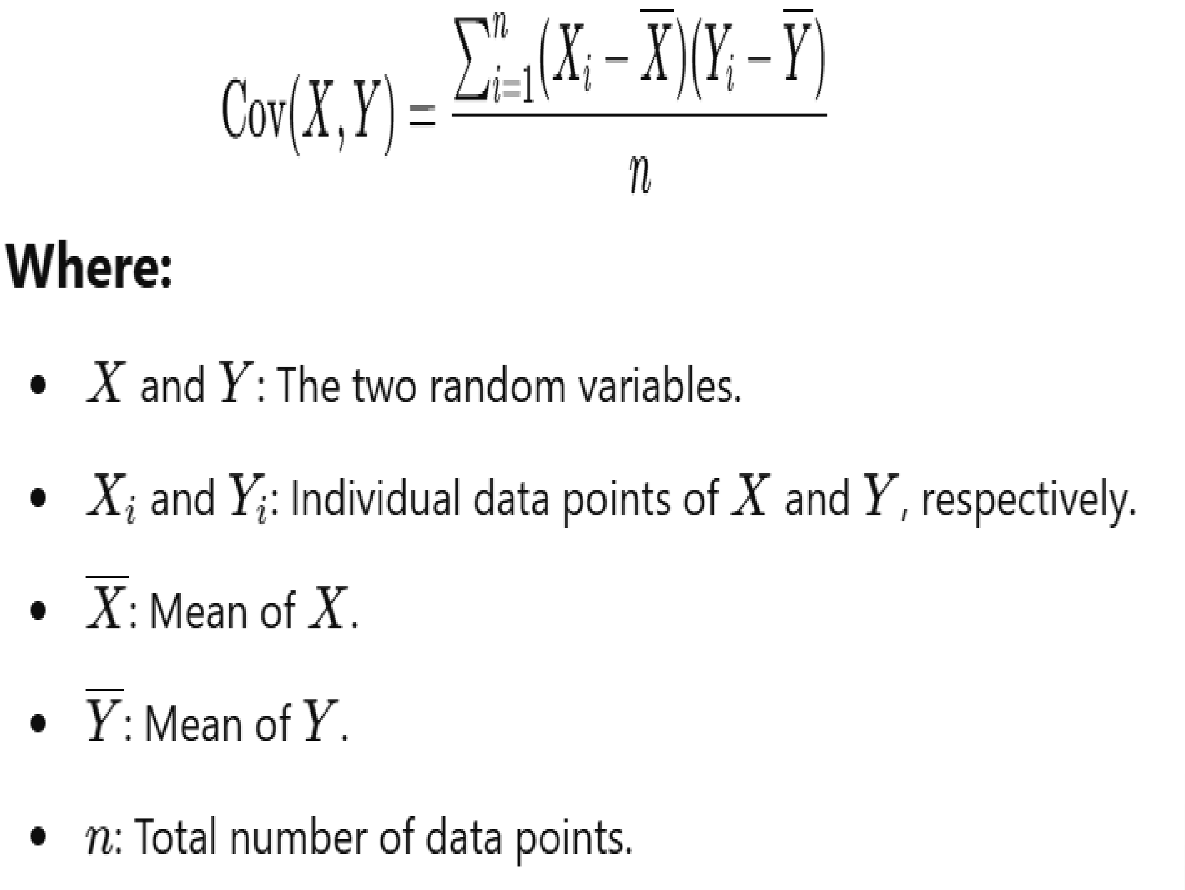
$$\text{Variance} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

- **Standard Deviation (SD):** The square root of variance, it gives a sense of the average distance of data points from the mean.

$$\text{Standard Deviation} = \sqrt{\text{Variance}}$$

Statistics for Machine Learning (Cont..)

- **Covariance:** Measures the relationship between two variables. Positive covariance means both variables increase together, while negative covariance means one increases as the other decreases



$$\mathrm{Cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{n}$$

Where:

- X and Y: The two random variables.
- X_i and Y_i: Individual data points of X and Y, respectively.
- $\overline{X}$: Mean of X.
- $\overline{Y}$: Mean of Y.
- n: Total number of data points.

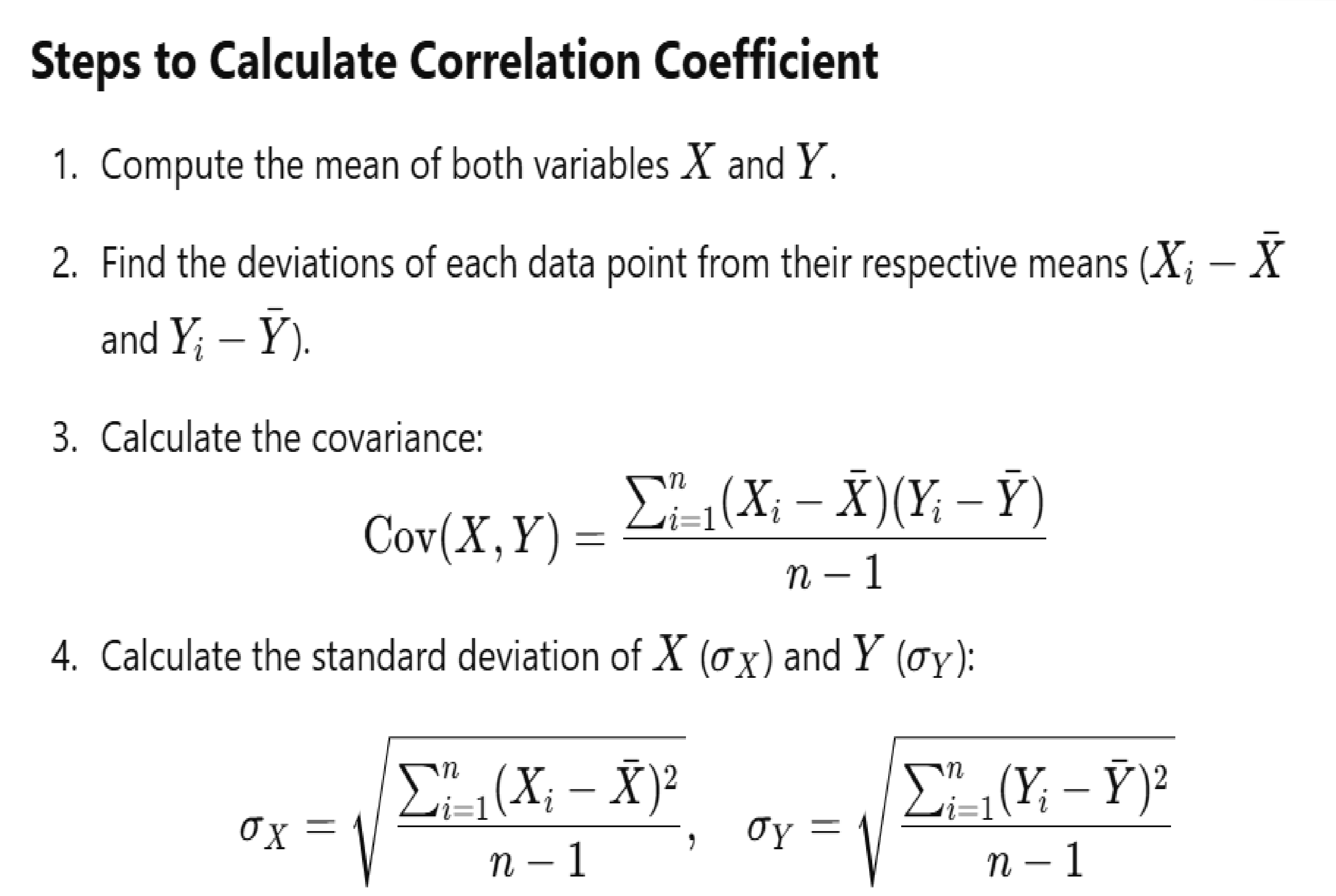
Statistics for Machine Learning (Cont..)

- **Correlation** is a statistical measure that quantifies the strength and direction of the relationship between two variables. It helps determine whether, and how strongly, two variables are related.

Types of Correlation

- **Positive Correlation**: As one variable increases, the other also increases.
- **Negative Correlation**: As one variable increases, the other decreases.
- **No Correlation**: No relationship exists between the variables.

Statistics for Machine Learning (Cont..)



Steps to Calculate Correlation Coefficient

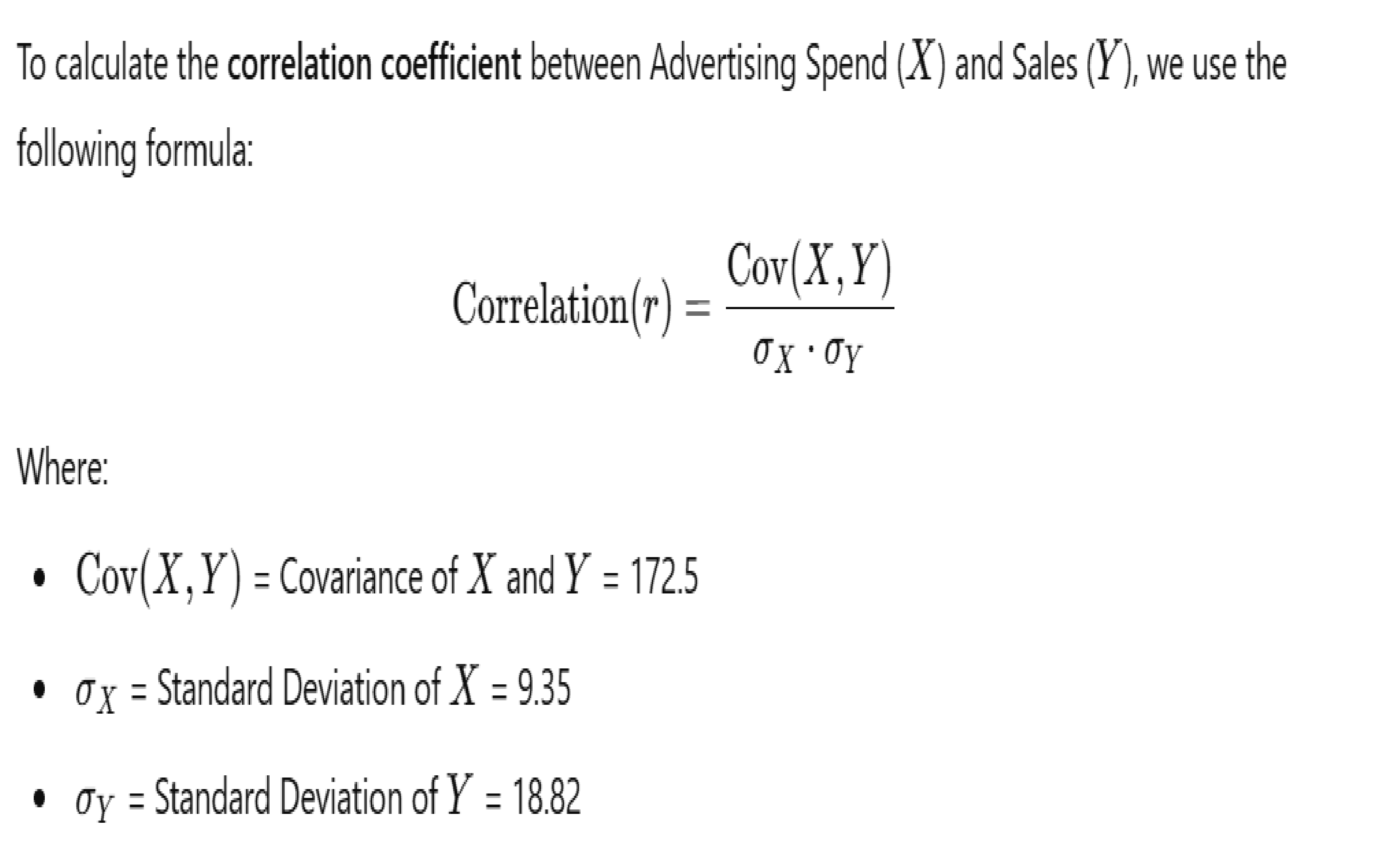
1. Compute the mean of both variables X and Y.
2. Find the deviations of each data point from their respective means ($X_i - \bar{X}$ and $Y_i - \bar{Y}$).
3. Calculate the covariance:

$$\text{Cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

4. Calculate the standard deviation of X (σ_X) and Y (σ_Y):

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}}, \quad \sigma_Y = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}{n-1}}$$

Statistics for Machine Learning (Cont..)



To calculate the **correlation coefficient** between Advertising Spend (X) and Sales (Y), we use the following formula:

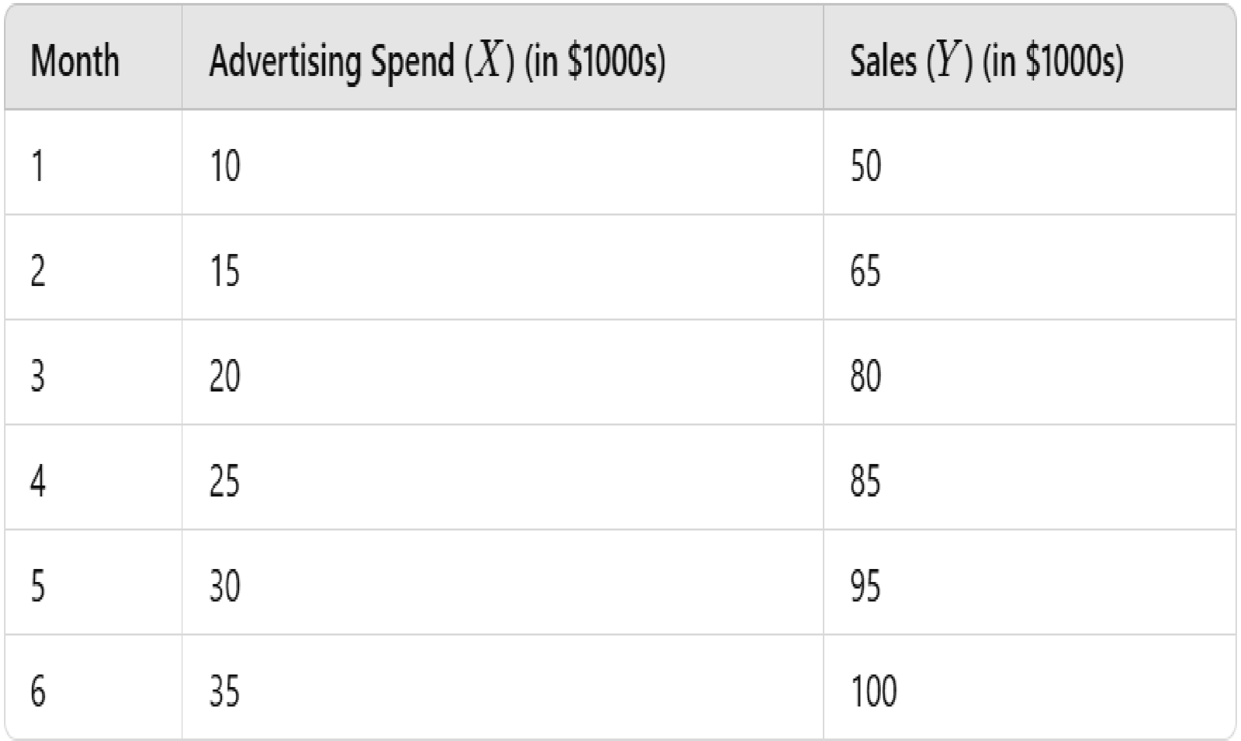
$$\text{Correlation}(r) = \frac{\text{Cov}(X,Y)}{\sigma_X \cdot \sigma_Y}$$

Where:

- $\text{Cov}(X,Y)$ = Covariance of X and Y = 172.5
- σ_X = Standard Deviation of X = 9.35
- σ_Y = Standard Deviation of Y = 18.82

Problem:

A retail company, BrightCo Retailers, wants to evaluate how its monthly advertising spend impacts monthly sales. They have collected the following data over six months



Month	Advertising Spend (X) (in \$1000s)	Sales (Y) (in \$1000s)
1	10	50
2	15	65
3	20	80
4	25	85
5	30	95
6	35	100

The company wants to answer the following questions:

- What is the average, median and mode for advertising spend and sales over six months?
- What is the variance and standard deviation for advertising spend and sales?
- Is there a relationship between advertising spend and sales? If so, is it positive or negative?
- Calculate the correlation coefficient to determine the strength of the relationship.

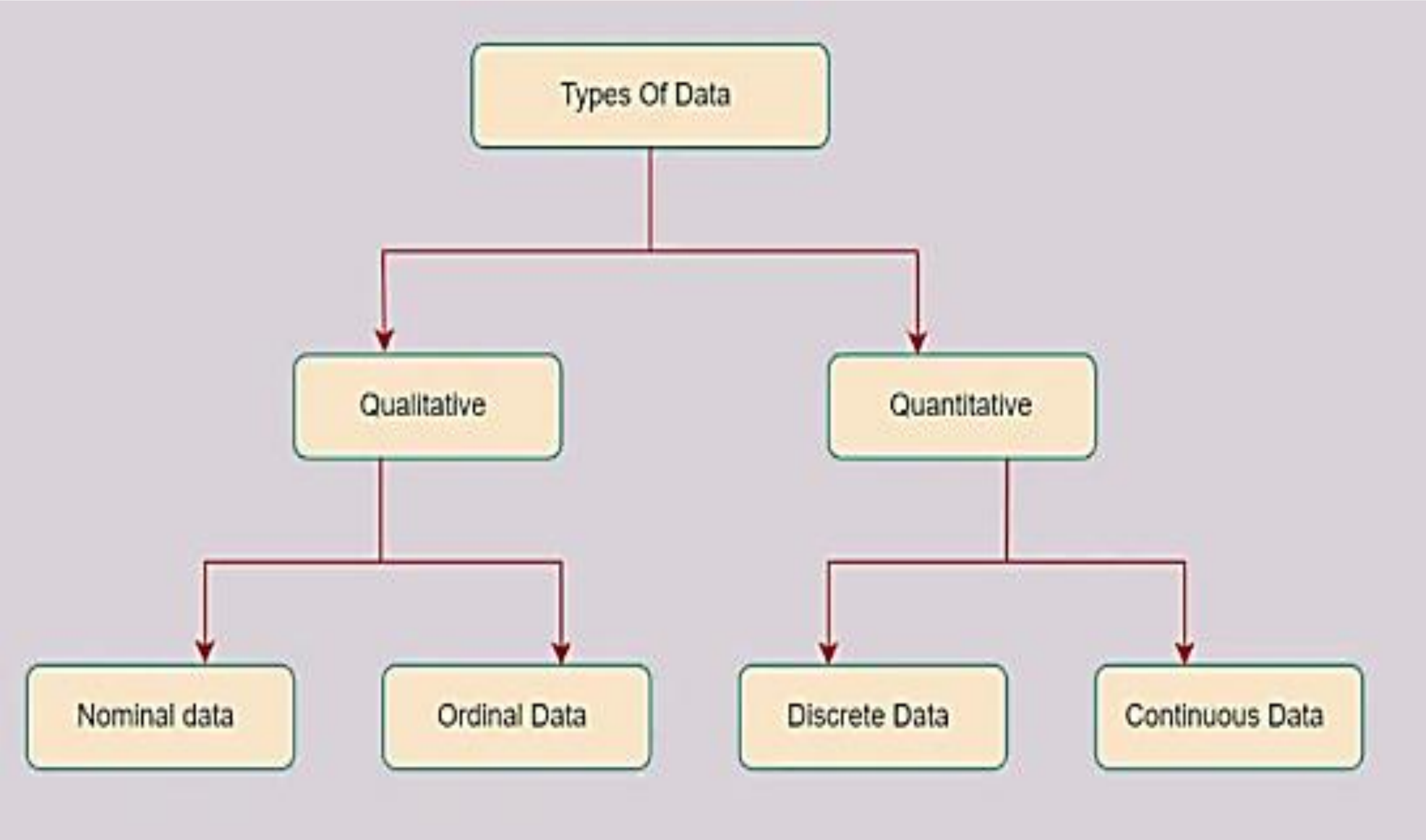
Turning Data into Probabilities

- Turning data into probabilities play a fundamental role in Machine Learning, especially in modeling uncertainty and making predictions.

Before calculating probabilities, need to understand the data:

- **Type of Data:** Determine if the data is categorical, ordinal, or continuous.
- **Range of Values:** Identify possible outcomes for each variable.
- **Distribution:** Analyse how the data is distributed (uniform, normal, exponential, etc.)

Types of Data:



Qualitative or Categorical Data is a type of data that can’t be measured or counted in the form of numbers

Examples of Nominal Data :

- Colour of hair (Blonde, red, Brown, Black, etc.)
- Marital status (Single, Widowed, Married)
- Nationality (Indian, German, American)
- Gender (Male, Female, Others)
- Eye Color (Black, Brown, etc.)

Examples of Ordinal Data :

- When companies ask for feedback, experience, or satisfaction on a scale of 1 to 10
- Letter grades in the exam (A, B, C, D, etc.)
- Ranking of people in a competition (First, Second, Third, etc.)
- Economic Status (High, Medium, and Low)

Quantitative data is a type of data that can be expressed in numerical values, making it countable and including statistical data analysis.

Examples of Discrete Data :

- Total numbers of students present in a class
- Cost of a cell phone
- Numbers of employees in a company
- The total number of players who participated in a competition
- Days in a week

Examples of Continuous Data :

- Height of a person
- Speed of a vehicle
- “Time-taken” to finish the work
- Wi-Fi Frequency
- Market share price

Four key approaches to turning data into probabilities:

1. Logistic Regression: Converting Inputs into Probabilities

- Logistic Regression is a statistical method used for **binary classification problems**. It transforms input features into probabilities using the **Sigmoid function**.

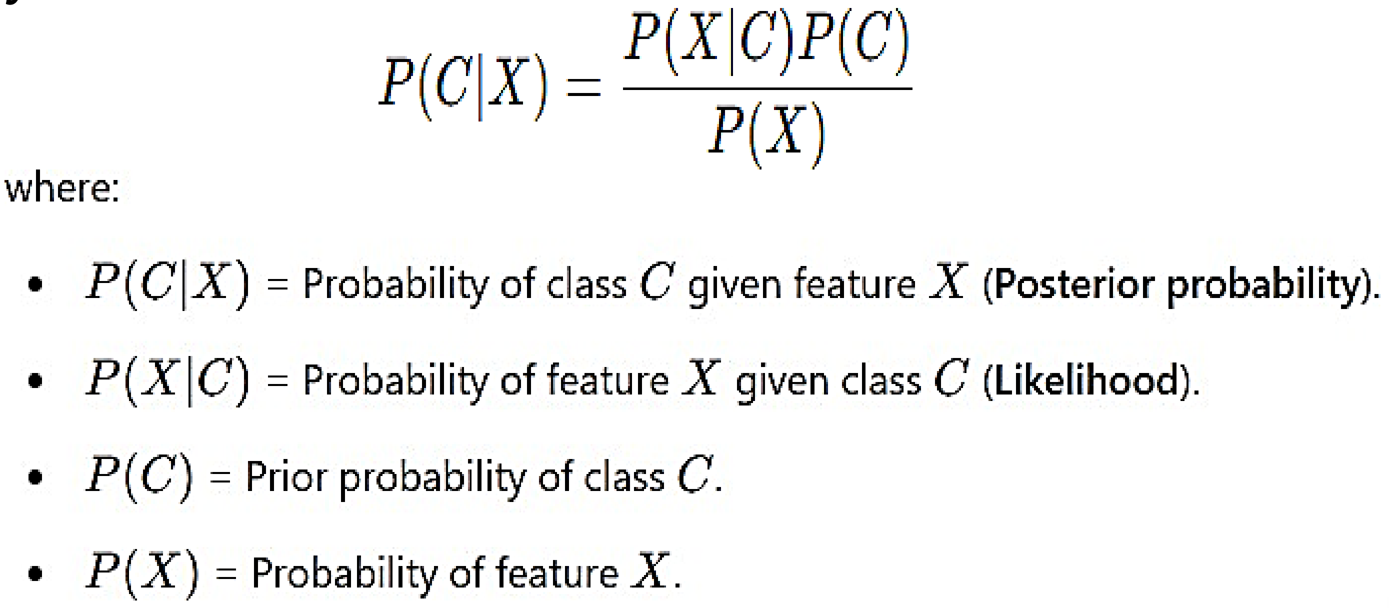
$$P(y = 1|X) = \frac{1}{1 + e^{-z}}$$

- **Example:**

$$P(\text{Diabetes}|\text{Age, BMI, Blood Sugar}) = \frac{1}{1 + e^{-(w_0 + w_1 \cdot \text{Age} + w_2 \cdot \text{BMI} + w_3 \cdot \text{Blood Sugar})}}$$

2. Naïve Bayes Classifier: Computing Probabilities Using Bayes' Theorem

- Naïve Bayes is a classification algorithm based on **Bayes' Theorem**, which computes the probability of a class given observed features.
- **Bayes' Theorem**



$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

where:

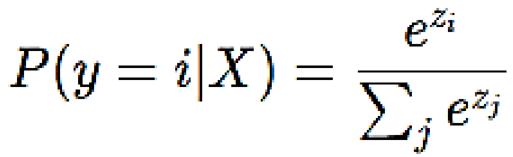
- $P(C|X)$ = Probability of class C given feature X **(Posterior probability)**.
- $P(X|C)$ = Probability of feature X given class C **(Likelihood)**.
- $P(C)$ = Prior probability of class C.
- $P(X)$ = Probability of feature X.

Example: Email Spam Detection

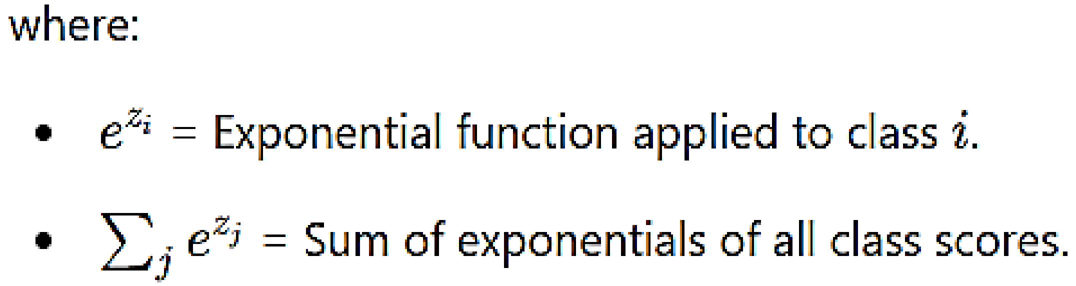
- A **spam filter** wants to classify emails as **Spam or Not Spam** based on words like "Free", "Money", and "Win"

3. Softmax Function: Assigning Probabilities in Multi-Class Classification

- The **Softmax function** is used in **multi-class classification** to assign probability values to multiple categories.



$$P(y = i|X) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$



where:

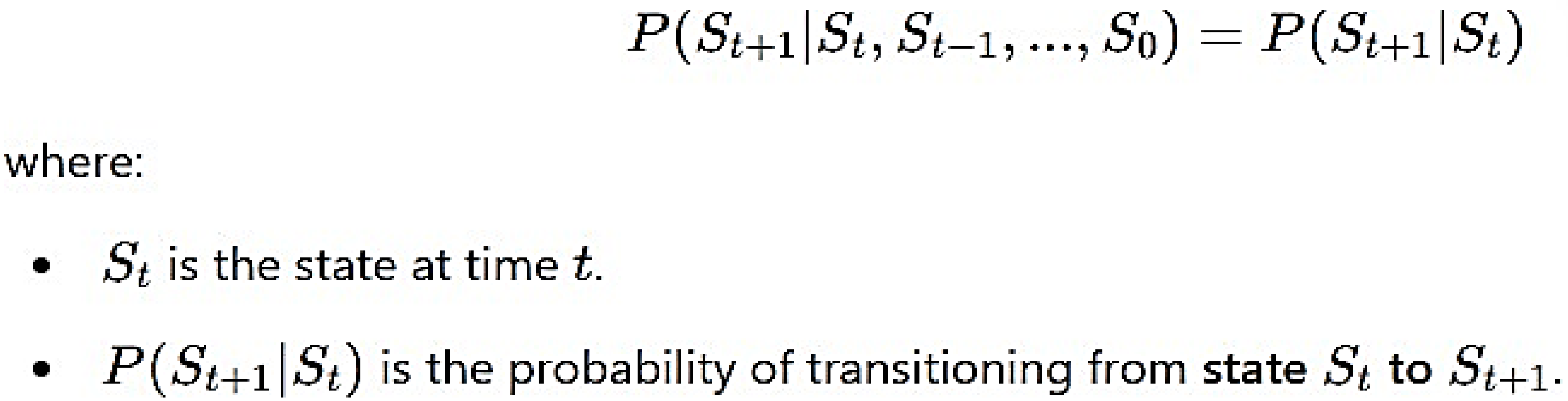
- e^{z_i} = Exponential function applied to class i.
- $\sum_j e^{z_j}$ = Sum of exponentials of all class scores.

Example: Image Classification (Handwritten Digit Recognition)

  - P(0)=0.01 P(1)=0.05 P(2)=0.02 P(9)=0.85
- Since **P(9) is the highest (85%)**, the image is classified as **Digit 9**.

4. Markov Models: Predicting Future Events Using Probabilities

- A **Markov Model** is a statistical model used to predict future events based on **probability transitions** between states.



$$P(S_{t+1}|S_t, S_{t-1}, ..., S_0) = P(S_{t+1}|S_t)$$

where:

- S_t is the state at time t.
- $P(S_{t+1}|S_t)$ is the probability of transitioning from **state** S_t **to** S_{t+1}.

Example: Weather Prediction

- A **weather model** predicts tomorrow’s weather based on today’s weather:

Today	**Tomorrow (Probability)**
Sunny	80% Sunny, 20% Rainy
Rainy	50% Sunny, 50% Rainy

Probability Theory

- **Probability** can be defined as the possibility of occurrence of an event. Probability is the likelihood or the chances that an uncertain event will occur.
- The probability of an event always lies between 0 and 1.
- ***Probability can be calculated by the number of times the event occurs divided by the total number of possible outcomes***
- P (H) = Number of ways to head occur/ total number of possible outcomes
- P (H) = ½
- P (H) = 0.5
- Where;
- P (H) = Probability of occurring Head as outcome while tossing a coin.

Probability Theory (Cont.)

- **Sample space:** The sample space is the collection of all potential outcomes of an experiment. For example, the sample space of flipping a coin is {heads, tails}.

- **Event:** An event is a collection of outcomes within the sample space. For example, the event of flipping a head is {heads}**.**

- **Probability:** The probability of an event is a number between 0 and 1 that represents the likelihood of the event occurring.

Types of Probability

Joint Probability: It tells the Probability of simultaneously occurring two random events.

- $P(A \cap B) = P(A). P(B)$

Where;

$P(A \cap B)$ = Probability of occurring events A and B both.

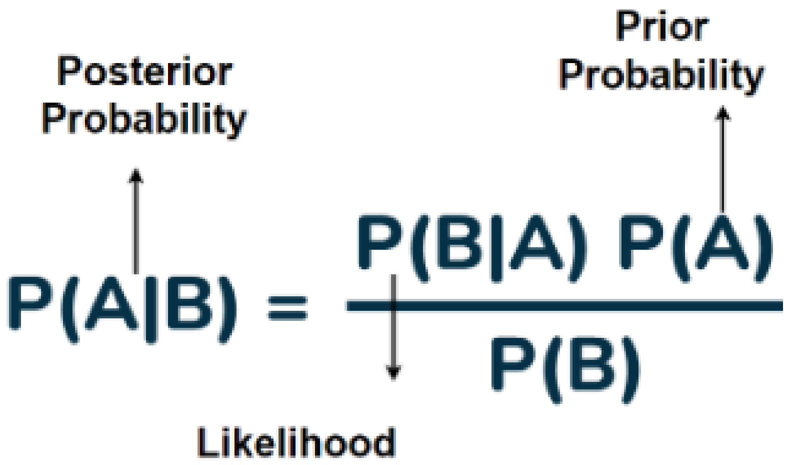
- P (A) = Probability of event A
- P (B) = Probability of event B

Conditional Probability: It is given by the Probability of event A given that event B occurred.

The Probability of an event A conditioned on an event B is denoted and defined as;

- $P(A | B) = P(A \cap B)/P(B)$

Bayes' Theorem Formula



Prior Probability

Posterior Probability

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Likelihood

- **Posterior probability (P(A|B)):**

  The probability of an event (A) occurring given that another event (B) has already occurred; essentially, the updated belief after seeing new data.
- **Prior probability (P(A)):**

  The initial belief about the probability of an event (A) before any new data is observed.

Bayes' Theorem Formula

- **Likelihood (P(B|A)):**

  The probability of observing data (B) given a specific parameter value (A).

- **Marginal probability (P(B)):**

  The probability of observing the data (B) across all possible parameter values, calculated by integrating the likelihood over the prior distribution.

Example:

- A bank wants to predict whether a customer will default on a loan given that they have a low credit score.
  - Let P(Default) = 0.3 (30% of customers default).
  - Let P(Low Credit Score) = 0.4 (40% of customers have low credit scores).
  - Let P(Low Credit Score | Default) = 0.8 (80% of defaulting customers had low credit scores).
- Using Bayes' Theorem, we calculate:

$$P(\text{Default}|\text{Low Credit Score}) = \frac{P(\text{Low Credit Score}|\text{Default}) \times P(\text{Default})}{P(\text{Low Credit Score})}$$

$$P(\text{Default}|\text{Low Credit Score}) = \frac{0.8 \times 0.3}{0.4} = \frac{0.24}{0.4} = 0.6$$

- So, given that a customer has a low credit score, the probability that they will default on a loan is 60%.

Probability Distributions

- In machine learning, **probability distributions** are fundamental for modeling uncertainty, making predictions, and understanding data patterns.
- They describe how likely different outcomes are in a dataset and are used in various applications such as generative models, Bayesian inference, and probabilistic classifiers.

Types of Probability Distributions in Machine Learning

1. Discrete Probability Distributions

- Used when the variable takes distinct values (e.g., categorical or count data).
- **Bernoulli Distribution**: Models a binary outcome (success/failure, 0/1).

$$P(X = x) = p^x(1 - p)^{1-x}, \quad x \in \{0, 1\}$$

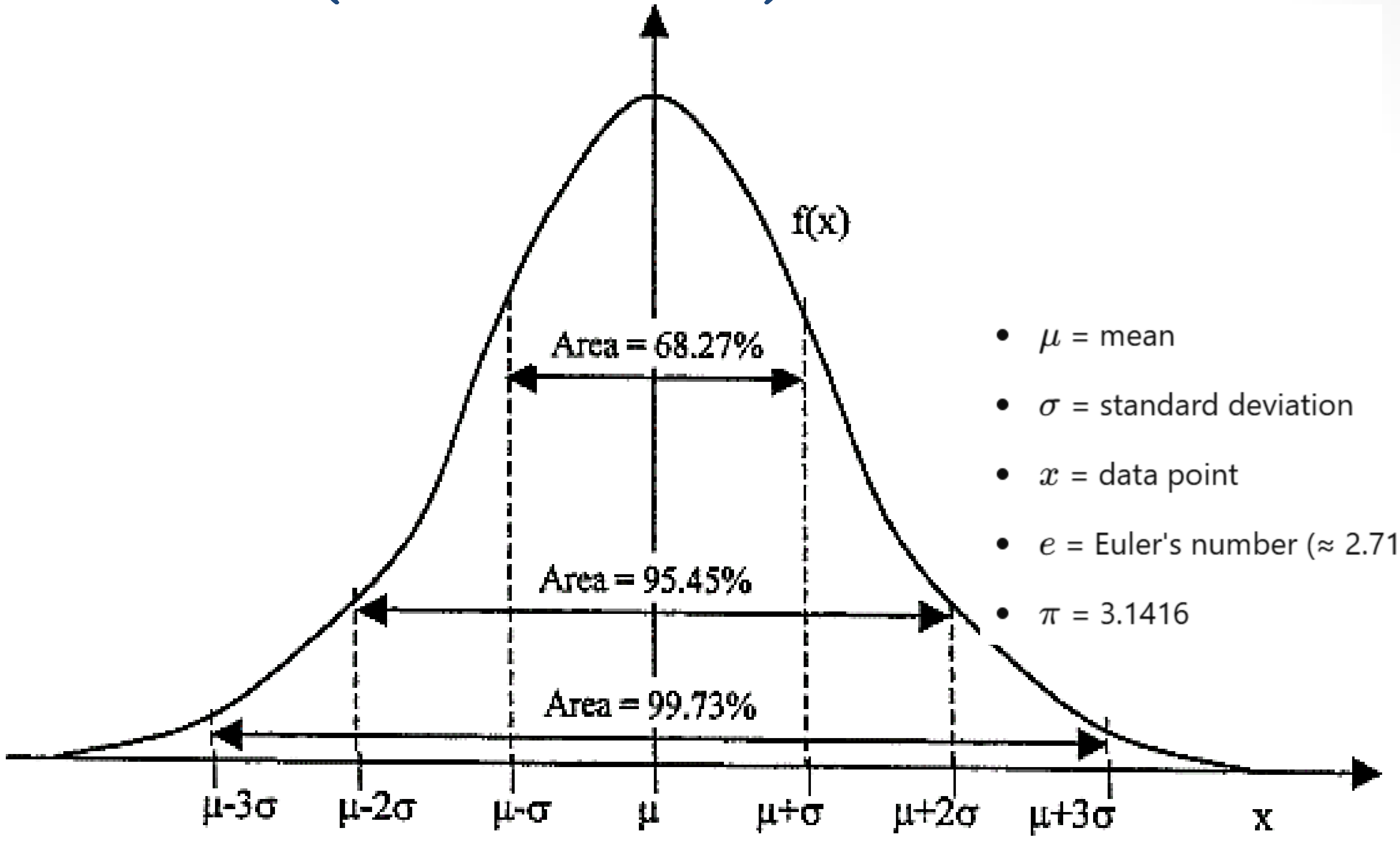
Probability Distributions

- **Binomial Distribution**: Describes the number of successes in a fixed number of trials (success or failure).
- **Poisson Distribution**: Describes the number of events occurring within a fixed interval of time or space (5 calls per hour but actually 3 calls occurred)

2. Continuous Probability Distributions

- Used for continuous variables (e.g., heights, temperatures).
- **Normal (Gaussian) Distribution**: A symmetric, bell-shaped curve where most of the data points are around the mean.
- In machine learning, many algorithms assume data is normally distributed.

Normal (Gaussian) Distribution:



- μ = mean
- σ = standard deviation
- x = data point
- e = Euler's number (≈ 2.718)
- π = 3.1416

- The probability density function (PDF) is given by:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Probability Distributions (Cont..)

- **Exponential Distribution**: Often used in machine learning for time-based events, like waiting times or survival analysis.
- **Uniform Distribution**: All outcomes in a range have equal probability.

Distribution	Type of Data	Example in ML
Normal (Gaussian)	Continuous	Height prediction, Stock price modeling
Binomial	Discrete (Success/Failure)	Loan approval, Email spam classification
Poisson	Count-based (Rare Events)	Predicting server failures, Fraud detection

DECISION THEORY IN MACHINE LEARNING

- Decision Theory is a mathematical framework used in Machine Learning (ML) to **make optimal choices under uncertainty.**
- It provides a structured way to **minimize risk and maximize utility** when making decisions based on available data.
- Decision theory is widely applied in fields such as **finance, healthcare, artificial intelligence (AI), and robotics**.

1. Risk Minimization

- Risk Minimization is a technique that helps models decide between multiple possible outcomes by minimizing the expected loss (also known as the cost function or error function).

Expected Loss Function

- The risk associated with a decision is evaluated using an Expected Loss Function:

$$R(a) = \sum_i L(a, y_i) P(y_i)$$

- Where:
  - R(a)= Expected risk of choosing action a.
  - $L(a,y_i)$= Loss incurred by taking action a when the actual outcome is y_i.
  - $P(y_i)$= Probability of the outcome y_i.

1. Risk Minimization (Cont..)

Applications in ML

✅ **Classification Problems:**

- In a spam detection model, misclassifying an important email as spam is more costly than missing a spam email. The model aims to minimize this loss.

✅ **Medical Diagnosis Models:**

- Predicting a disease incorrectly can have different risks.
  - False Negative (FN): A patient has cancer, but the model predicts no cancer (high risk).
  - False Positive (FP): A patient does not have cancer, but the model predicts cancer (less risky).

- The ML model is trained to minimize the expected loss associated with incorrect predictions.

2. Bayesian Decision Theory

- Bayesian Decision Theory is an extension of probability theory that helps ML models make optimal decisions using prior knowledge (probabilities).
- ***Bayes' Theorem***

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)}$$

- Where:
  - P(H|D) = Probability of hypothesis H given data D (Posterior Probability).
  - P(D|H) = Probability of data D given hypothesis H (Likelihood).
  - P(H) = Prior probability of hypothesis H (Prior Knowledge).
  - P(D) = Probability of the data occurring (Evidence).

2. Bayesian Decision Theory (Cont..)

Bayesian Decision Rule

- A decision is made based on posterior probabilities, and the action that minimizes risk is chosen:

$$a^* = \arg\min_a \sum_i L(a, y_i) P(y_i|x)$$

Where:

- a∗ = Optimal action that minimizes expected loss.
- P(yi|x) = Posterior probability of class yi given input x.

Applications in ML

- Naïve Bayes Classifier
- Face Recognition Systems
- Autonomous Robots
- Fraud Detection in Banks.

3. Utility Function

- A Utility Function measures the usefulness or benefit of a decision. While Risk Minimization focuses on reducing loss, a Utility Function focuses on maximizing gains or rewards.
- The utility of a decision a is given by:

$$U(a) = \sum_i P(y_i) \times U(a, y_i)$$

- Where:
  - U(a) = Utility of action a.
  - $P(y_i)$ = Probability of outcome y_i
  - U(a,yi) = Utility of taking action a when the actual outcome is yi

3. Utility Function (Cont..)

A bank evaluates loan applicants and assigns utility scores to different decisions:

- High-credit-score applicants have a higher utility because they are low-risk.
- Low-credit-score applicants have a lower utility due to higher default risk.

Decision	**Credit Score**	**Utility Score**
Approve Loan	High (750+)	+100
Approve Loan	Medium (600-750)	+50
Approve Loan	Low (<600)	-200 (high risk)
Reject Loan	Low (<600)	+50
Reject Loan	High (750+)	-50 (lost opportunity)

3. Utility Function (Cont..)

Applications in ML

- **Recommender Systems**
  - Netflix assigns a utility score to movies based on user preferences and suggests the most relevant content.
- **AI Chatbots (Customer Support)**
  - Chatbots maximize customer satisfaction by choosing responses that maximize utility
  - (e.g., solving a customer's issue quickly).
- **Stock Market Predictions**
  - An AI trading bot selects investments that maximize expected returns while minimizing risk

Comparison of Concepts

Concept	Purpose	Formula	Example in ML
Risk Minimization	Minimize expected loss	$R(a) = \sum_i L(a, y_i) P(y_i)$	Medical diagnosis, Self-driving cars
Bayesian Decision Theory	Make decisions using probability	$P(H\|D) = \frac{P(D\|H)P(H)}{P(D)}$	Spam or not spam, Fraud Detection in Bank, Face Recognition systems
Utility Function	Maximize benefits	$U(a) = \sum_i P(y_i) \times U(a, y_i)$	Loan approval, Recommender systems