CSE-503 MACHINE LEARNING

PARTHO SARATHI SARKER

ASST. PROFESSOR

DEPT. OF COMPUTER SCIENCE AND ENGINEERING(CSE)

UNIVERSITY OF GLOBAL VILLAGE (UGV)

MACHINE LEARNING

Course Code:	CSE 503	Credits:	03
		CIE Marks:	60
Exam Hours:	03	SEE Marks:	40

Course Learning Outcome (CLOs): After Completing this course successfully, the student will be able to…

CLO1	Understand fundamental concepts of machine learning, including its algorithms, datasets, and applications.
CLO2	Analyze and preprocess data for training machine learning models.
CLO3	Apply machine learning algorithms to solve classification, regression, and clustering problems.
CLO4	Evaluate and optimize machine learning models using performance metrics.
CLO5	Develop and implement machine learning models in practical scenarios using programming tools.

ASSESSMENT PATTERN

Bloom's Category Marks (out of 90)	Tests (30)	Assignments (10)	Quizzes (10)	Attendance (10)
Remember	04	03		
Understand	05	02	04	
Apply	06	03	03	
Analyze	05			
Evaluate	05	02	03	
Create	05			

SEE- Semester End Examination (40 Marks)

Bloom's Category	Test
Remember	05
Understand	05
Apply	13
Analyze	07
Evaluate	05
Create	05

COURSE PLAN

Week	Topics	Teaching Strategy	Assessment Strategy	Mapped CLO(s)
1	Introduction to Machine Learning	Lecture with real-world examples, discussion on ML applications.	Quiz, class participation.	CLO1
2	Types of Machine Learning	Lecture and examples of supervised, unsupervised, and reinforcement learning.	Assignment: Differentiate between ML types with examples.	CLO1
3	Data and Datasets in ML	Data visualization, discussions on dataset preparation, cleaning, and preprocessing.	Hands-on activity: preprocess a sample dataset.	CLO1, CLO2
4	Dataset Properties and Analysis	Interactive lecture on data properties, outliers, and probability distributions.	Assignment: Analyze properties of a given dataset.	CLO2
5	Supervised Learning: Regression	Explanation of linear and logistic regression, practical coding session.	Practical task: Implement and evaluate regression models.	CLO1, CLO3
6	Supervised Learning: KNN Algorithm	Visualization and demonstration of KNN, distance metrics, and hands-on implementation.	Quiz and coding exercise on KNN.	CLO3
7	Decision Tree Algorithm	Flowchart explanation of decision trees, hands-on activity using a dataset.	Assignment: Build and test a decision tree.	CLO3
8	Unsupervised Learning: Clustering	Explanation of clustering techniques, practical coding on k-means clustering.	Practical: Apply k-means clustering to a dataset.	CLO1, CLO3

COURSE PLAN

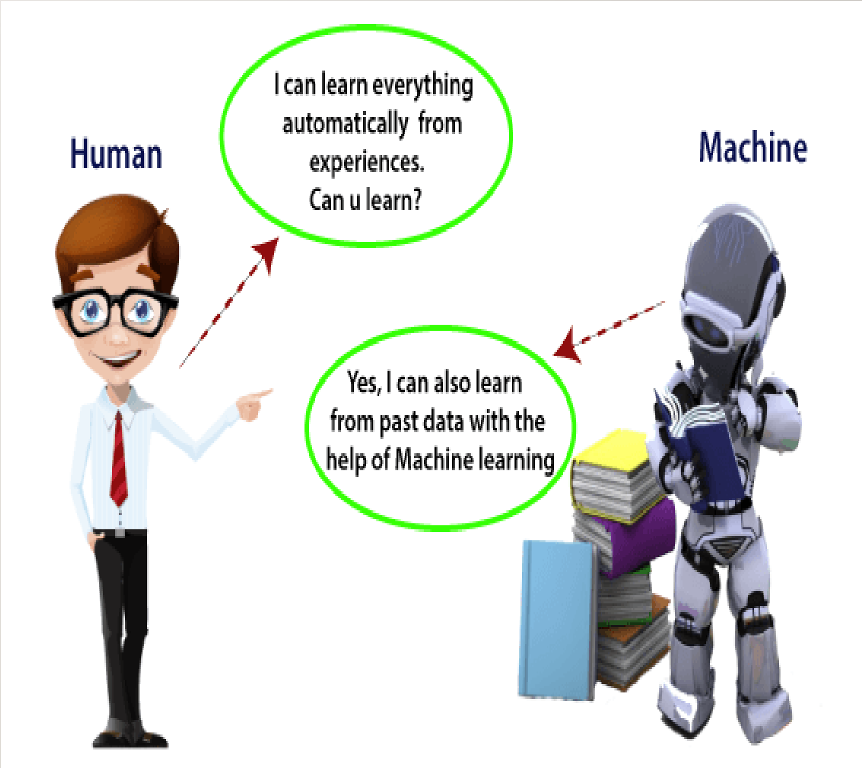
Week	Topics	Teaching Strategy	Assessment Strategy	Mapped CLO(s)
9	Implementation Strategies	Lecture, multimedia, discussions	Final term assessments	CLO4
10	Testing in Systems Design	Revision through Q&A, group activities	Participation, group evaluation	CLO4
11	Scalability and Performance	Lecture, multimedia, discussions	Feedback, Q&A, assessment	CLO4
12	Security in System Design	Lecture, discussions	Q&A, Midterm assessments	CLO4
13	Advanced Analysis Techniques	Lecture, multimedia	Midterm assessments	CLO2, CLO4
14	Challenges in System Design	Lecture, discussions	Midterm assessments	CLO4
15	Roles and Responsibilities	Lecture, discussions	Feedback, Q&A, assessment	CLO1, CLO2
16	Real-World Applications	Lecture, multimedia, discussions	Q&A, assignments	CLO5
17	Capstone Project and Review	Lecture, multimedia, discussions	Ethical analysis, Final assessments	CLO5

WEEK 1

MACHINE LEARNING, DIFFERENCE BETWEEN TRADITIONAL PROGRAMMING AND MACHINE LEARNING,

WHAT IS MACHINE LEARNING?

- Machine learning is a branch of artificial intelligence that enables algorithms to uncover hidden patterns within datasets, allowing them to make predictions on new, similar data without explicit programming for each task.



WHAT IS MACHINE LEARNING?

- For instance, recommender systems use historical data to personalize suggestions. Netflix, for example, employs collaborative and content-based filtering to recommend movies and TV shows based on user viewing history, ratings, and genre preferences. Reinforcement learning further enhances these systems by enabling agents to make decisions based on environmental feedback, continually refining recommendations.

DIFFERENCE BETWEEN TRADITIONAL PROGRAMMING AND MACHINE LEARNING

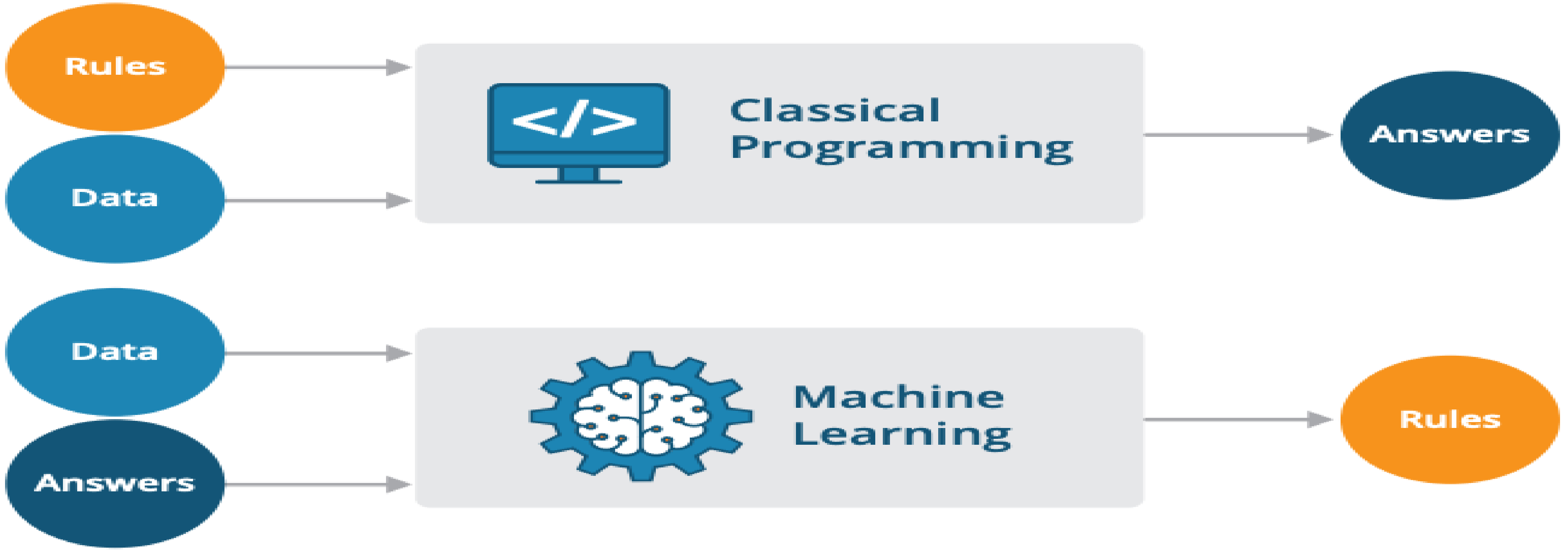


Figure 1. Classical programming vs machine learning

DIFFERENCE BETWEEN TRADITIONAL PROGRAMMING AND MACHINE LEARNING

Approach to Problem Solving:

1. **Traditional Programming**: In traditional programming, a programmer writes explicit rules or instructions for the computer to follow. These rules dictate exactly how the computer should process input data to produce the desired output. It requires a deep understanding of the problem and a clear way to encode the solution in a programming language.
2. **Machine Learning**: In machine learning, instead of writing explicit rules, a programmer trains a model using a large dataset. The model learns patterns and relationships from the data, enabling it to make predictions or decisions without being explicitly programmed for each possibility. This approach is particularly useful for complex problems where defining explicit rules is difficult or impossible.

Data Dependency:

1. **Traditional Programming**: Relies less on data. The quality of the output depends mainly on the logic defined by the programmer.
2. **Machine Learning**: Heavily reliant on data. The quality and quantity of the training data significantly impact the performance and accuracy of the model.

Flexibility and Adaptability:

1. **Traditional Programming**: Has limited flexibility. Changes in the problem domain require manual updates to the code.
2. **Machine Learning**: Offers higher adaptability to new scenarios, especially if the model is retrained with updated data.

DIFFERENCE BETWEEN TRADITIONAL PROGRAMMING AND MACHINE LEARNING

Problem Complexity:

1. **Traditional Programming**: Best suited for problems with clear, deterministic logic.
2. **Machine Learning**: Better for dealing with complex problems where patterns and relationships are not evident, such as image recognition, natural language processing, or predictive analytics. Learn about What is predictive analytics.

Development Process:

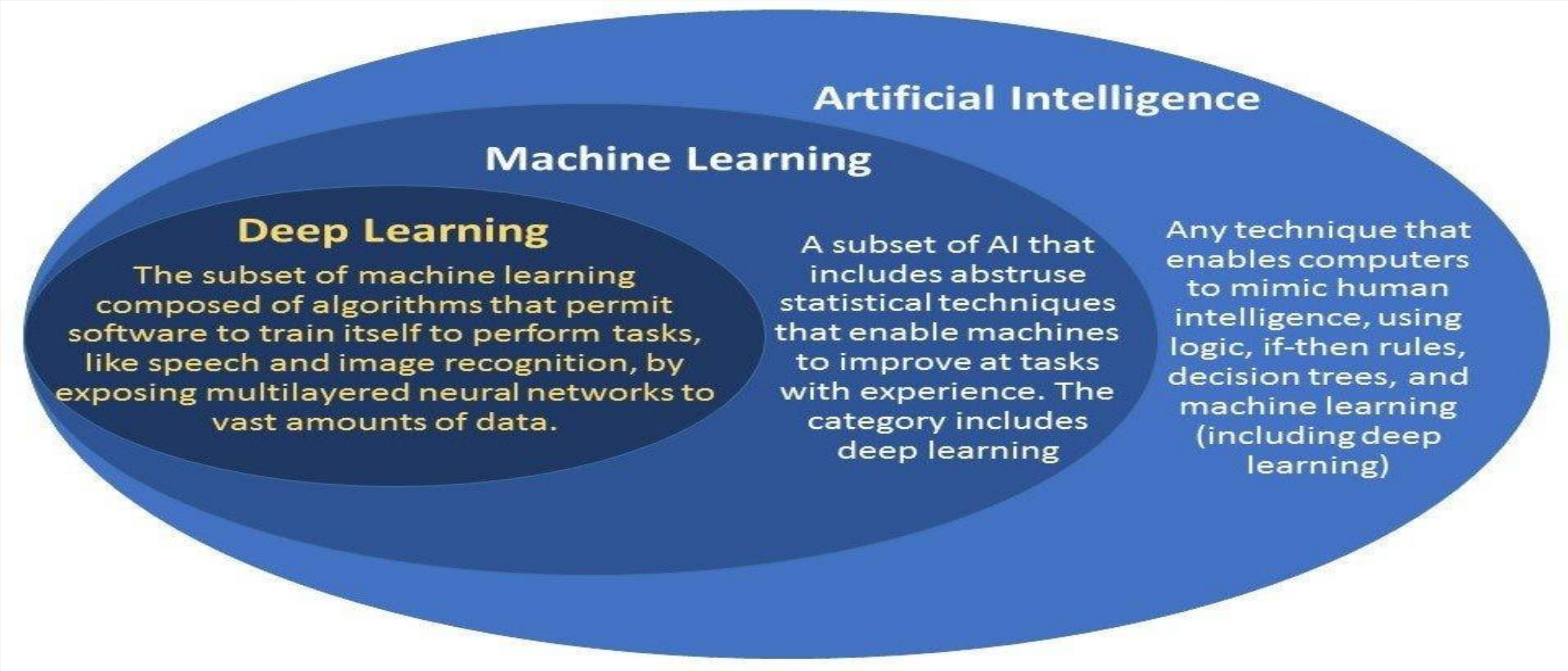
1. **Traditional Programming**: The development process is generally linear and predictable, focusing on implementing and debugging predefined logic.
2. **Machine Learning**: Involves an iterative process where models are trained, evaluated, and fine-tuned. This process can be less predictable and more experimental.

Outcome Predictability:

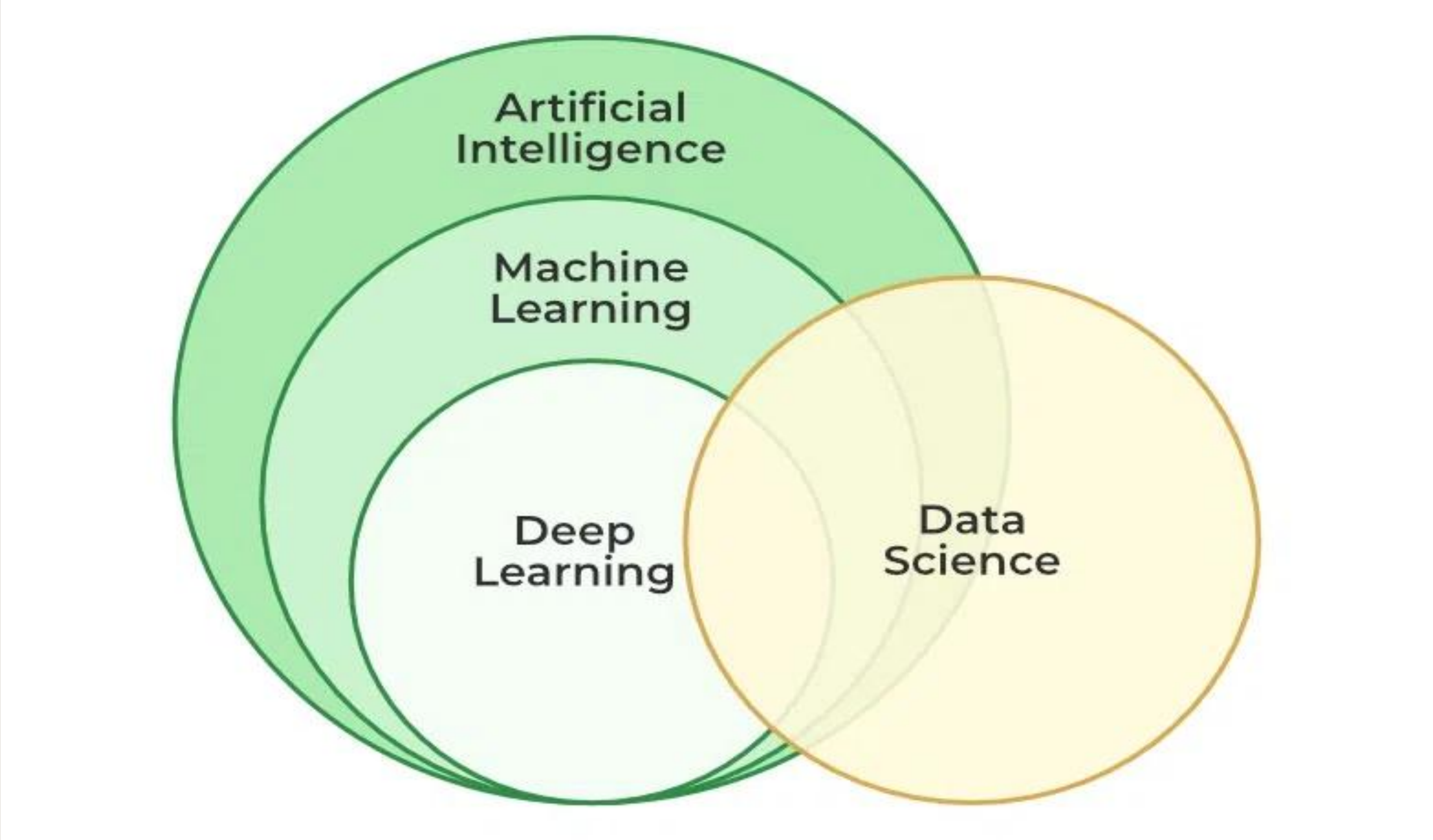
1. **Traditional Programming**: The outcome is highly predictable if the inputs and the logic are known.
2. **Machine Learning**: Predictions or decisions made by a machine learning model can sometimes be less interpretable, especially with complex models like deep neural networks.

WEEK 2
RELATION AMONG AI, ML, DEEP LEARNING, DIFFERENCE BTW AI, ML, & DS

RELATION AMONG AI, ML, DEEP LEARNING

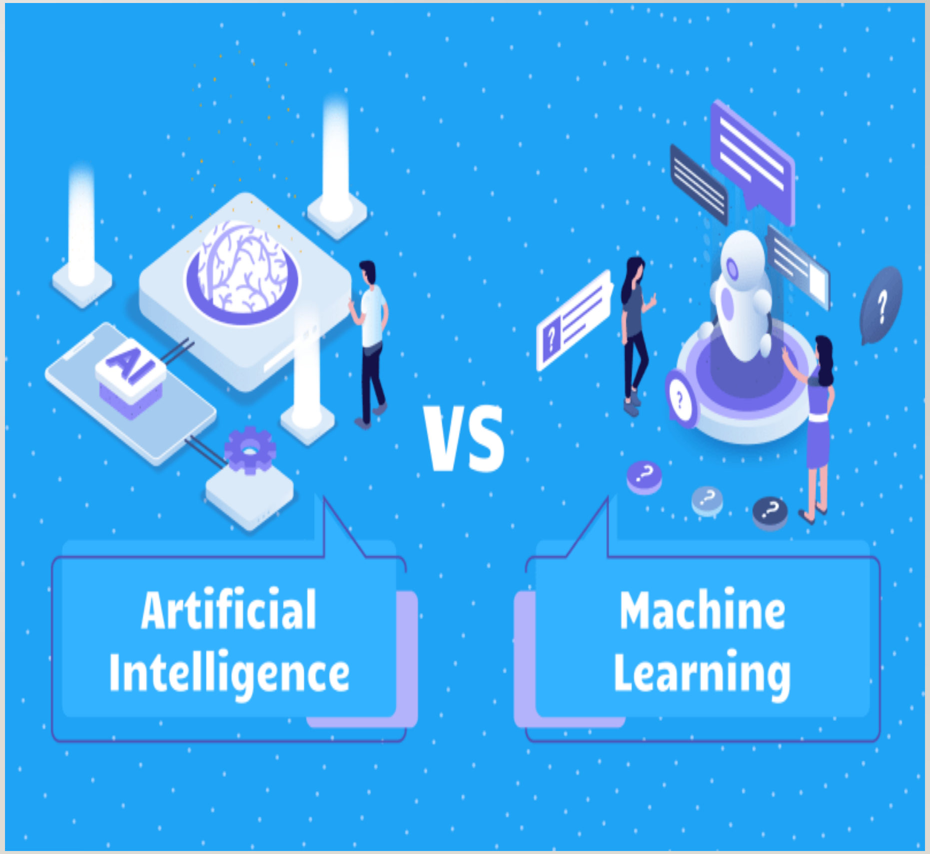


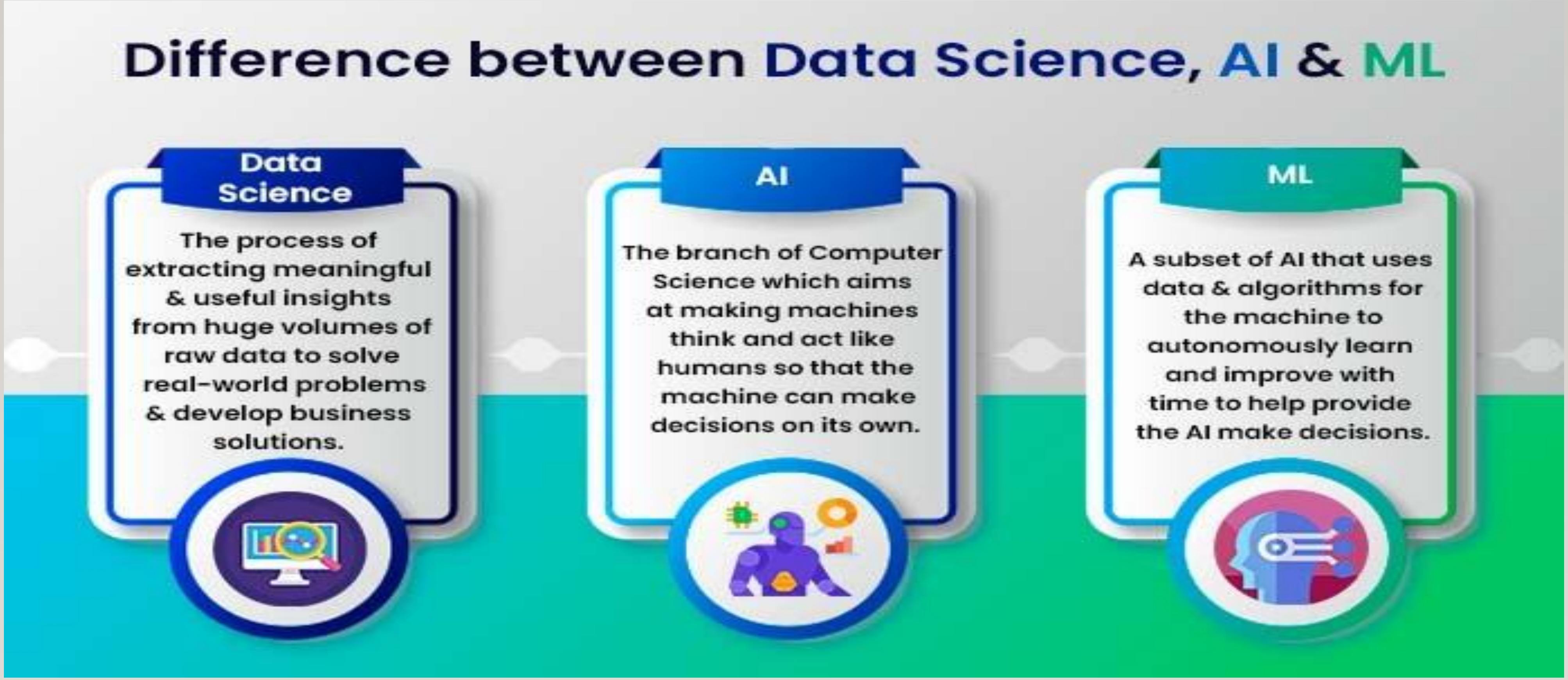
WHERE IS DATA SCIENCE???



DIFFERENCE BTW AI & ML

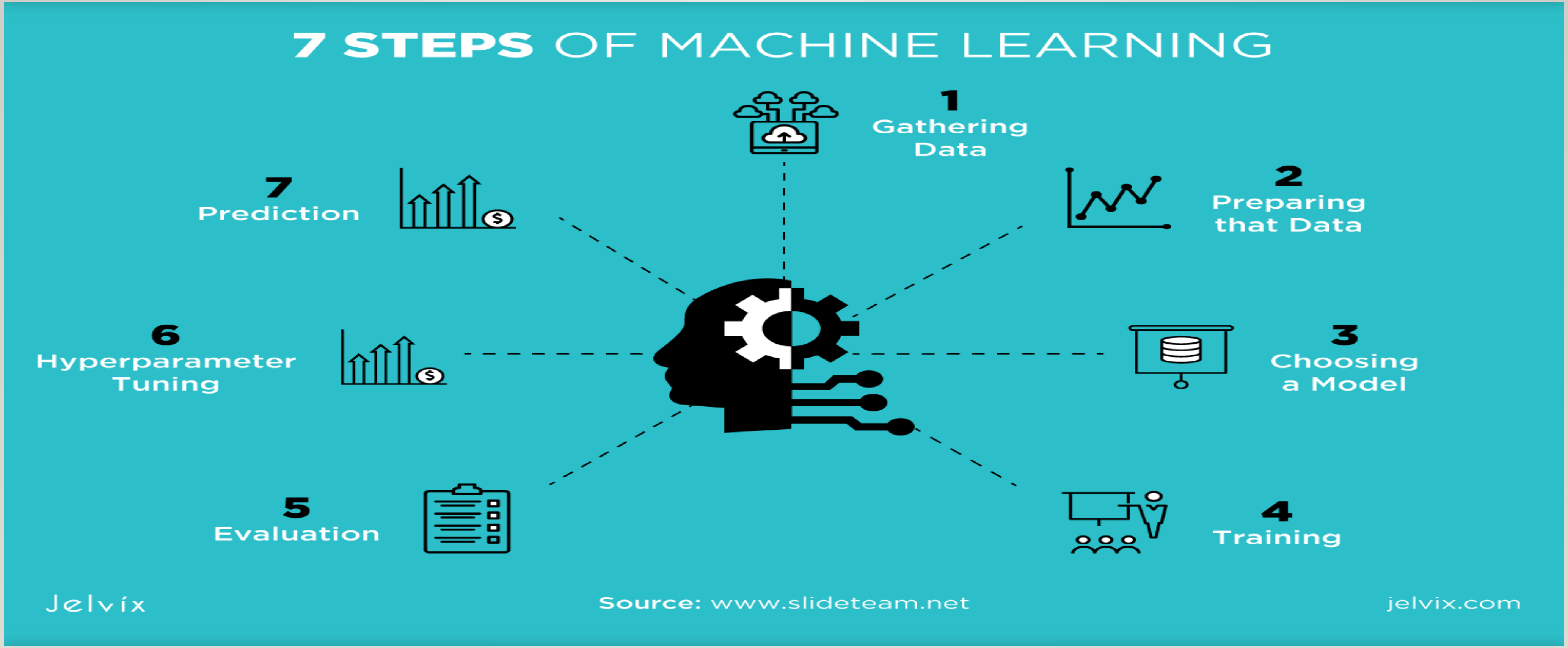
Artificial Intelligence	Machine Learning
Artificial intelligence (AI), where intelligence is defined as the acquisition of knowledge and the ability to apply knowledge.	Machine Learning (ML) means gaining skill or knowledge.
The goal is not accuracy but to increase the chance of business success.	The goal is to increase accuracy, but it does not care about business success
This leads to the development of a system that mimics a human being to behave in situations.	It involves designing self-learning algorithms.
The aim is to simulate natural intelligence to solve tough issues	The aim is to learn from the data on the specific task to maximize the performance of the machine.
Artificial Intelligence is a decision maker	ML enables the system to learn new things from the data.
It works as a smart working computer program	It is a simple concept machine that takes data and learns from data.
AI finds optimal solution	ML finds only solution, whether it is optimal or not.



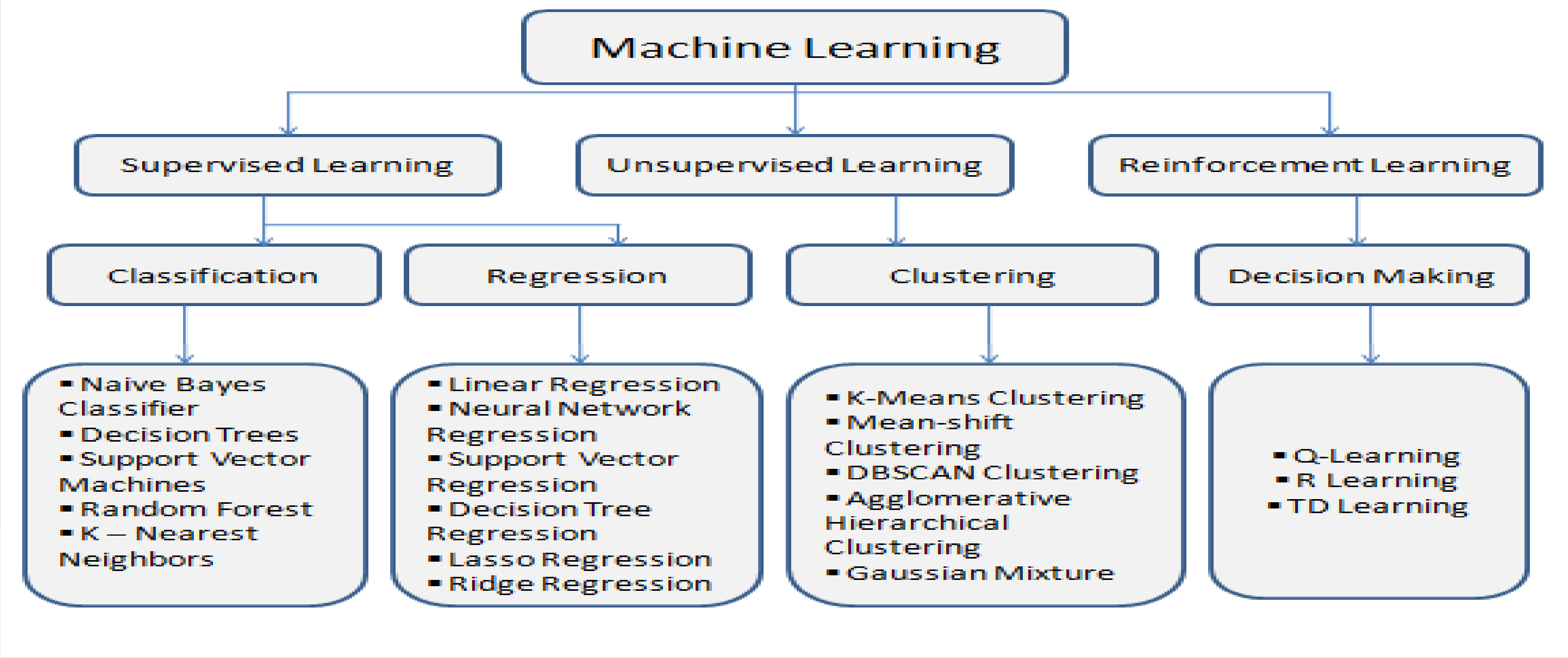

Difference between Data Science, AI & ML
Data Science
The process of extracting meaningful & useful insights from huge volumes of raw data to solve real-world problems & develop business solutions.
AI
The branch of Computer Science which aims at making machines think and act like humans so that the machine can make decisions on its own.
ML
A subset of AI that uses data & algorithms for the machine to autonomously learn and improve with time to help provide the AI make decisions.

WEEK 3
STEP OF ML ALGORITHM, TYPE OF MACHINE LEARNING ALGORITHM, TYPES OF DATASETS

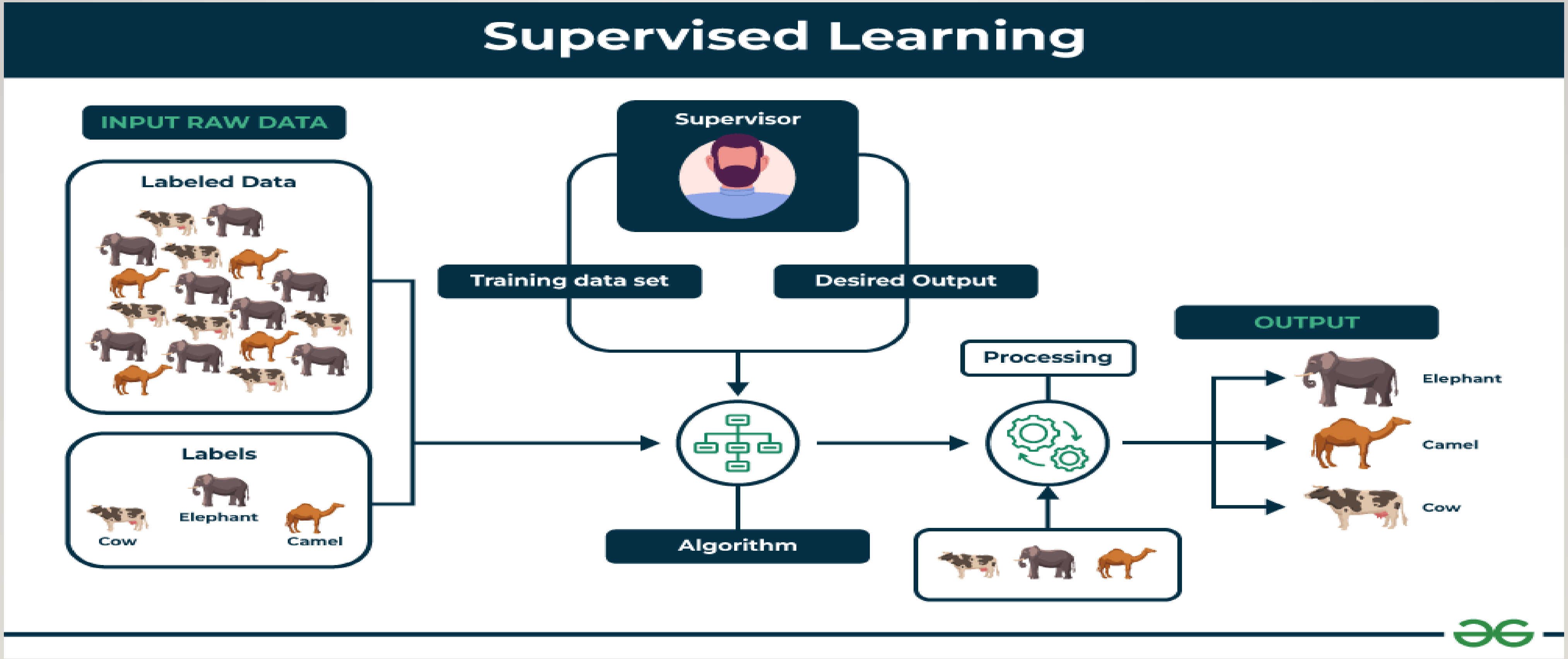
HOW TO APPLY ML ALGORITHM???



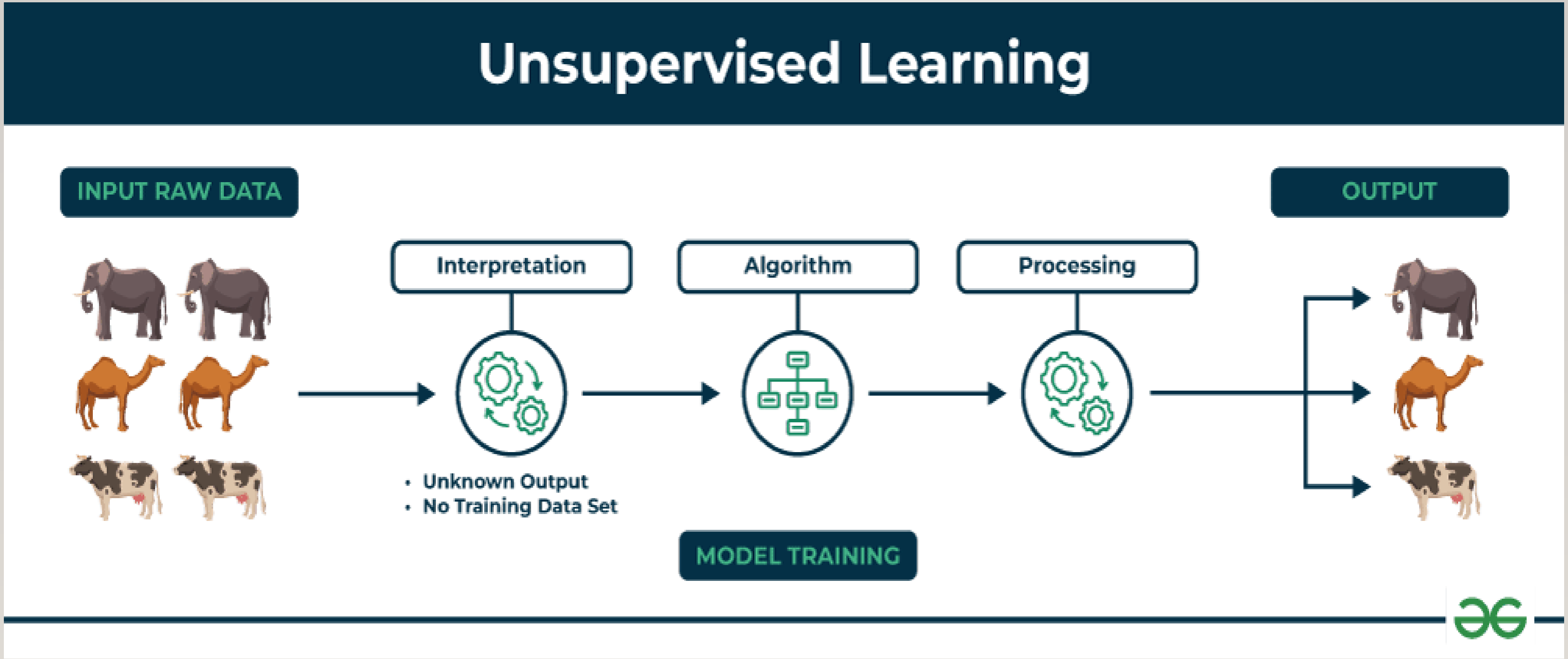
TYPE OF MACHINE LEARNING ALGORITHM.



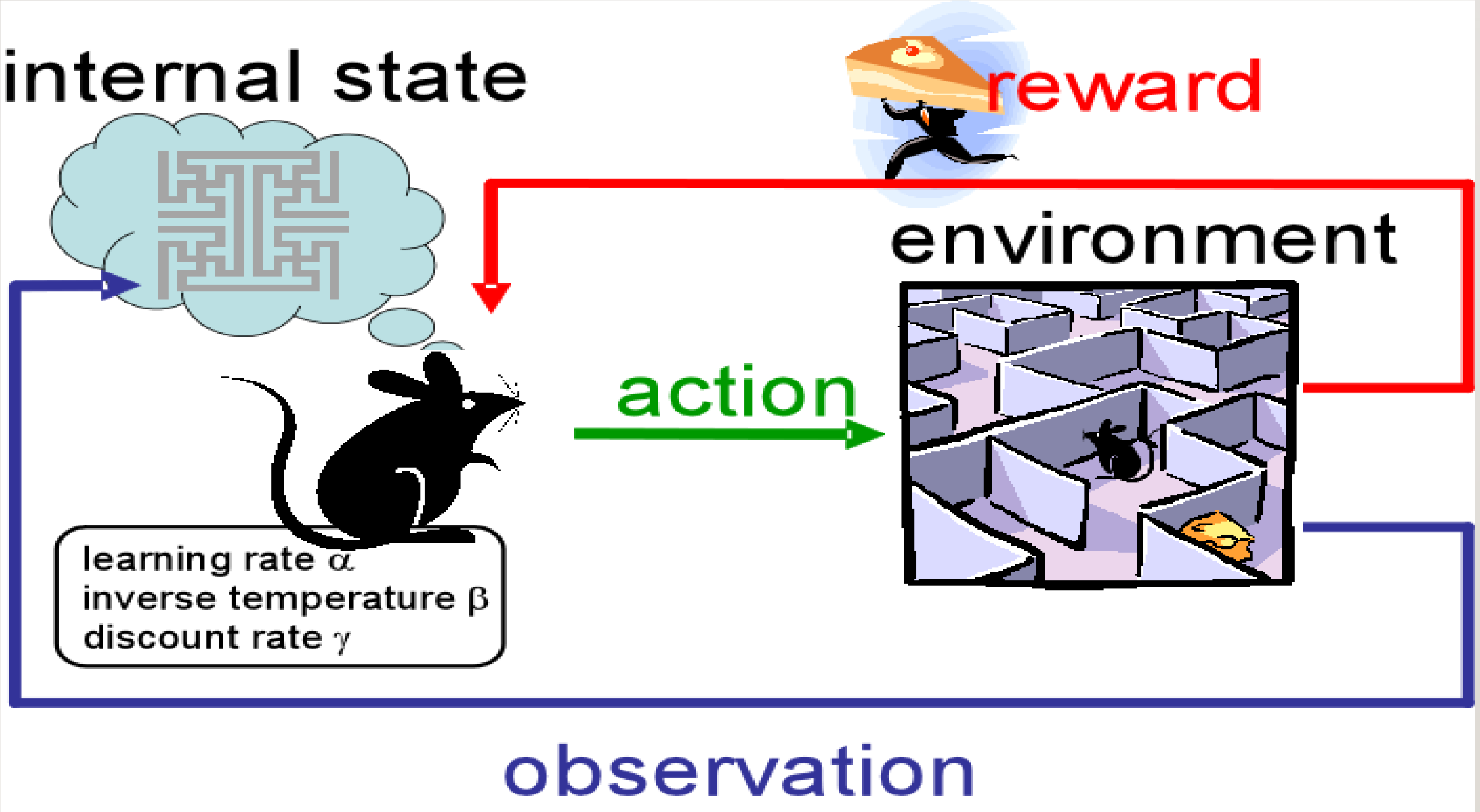
SUPERVISED LEARNING

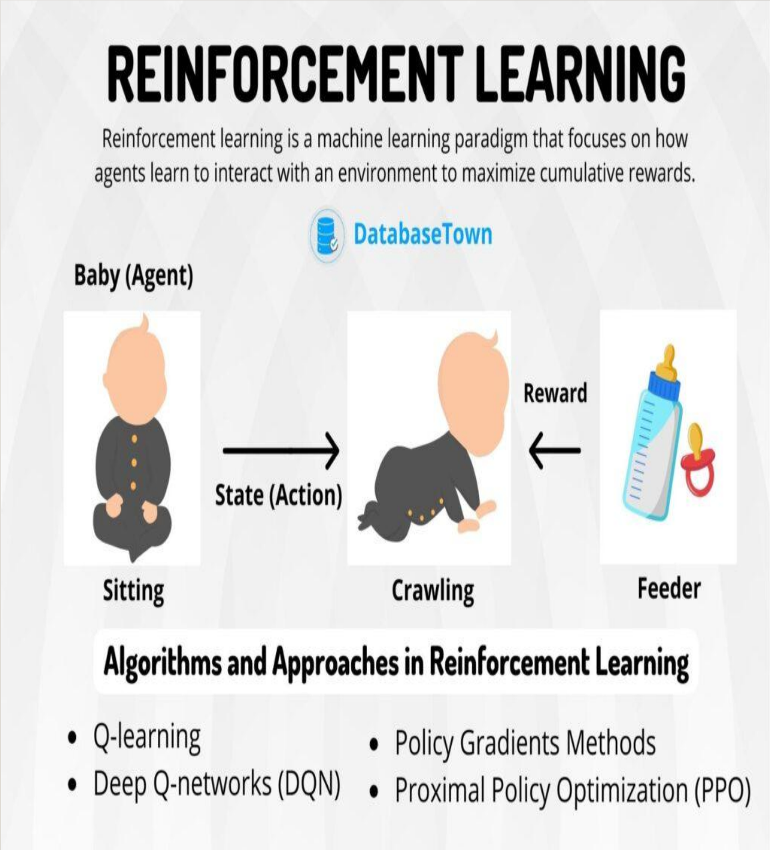


UNSUPERVISED LEARNING

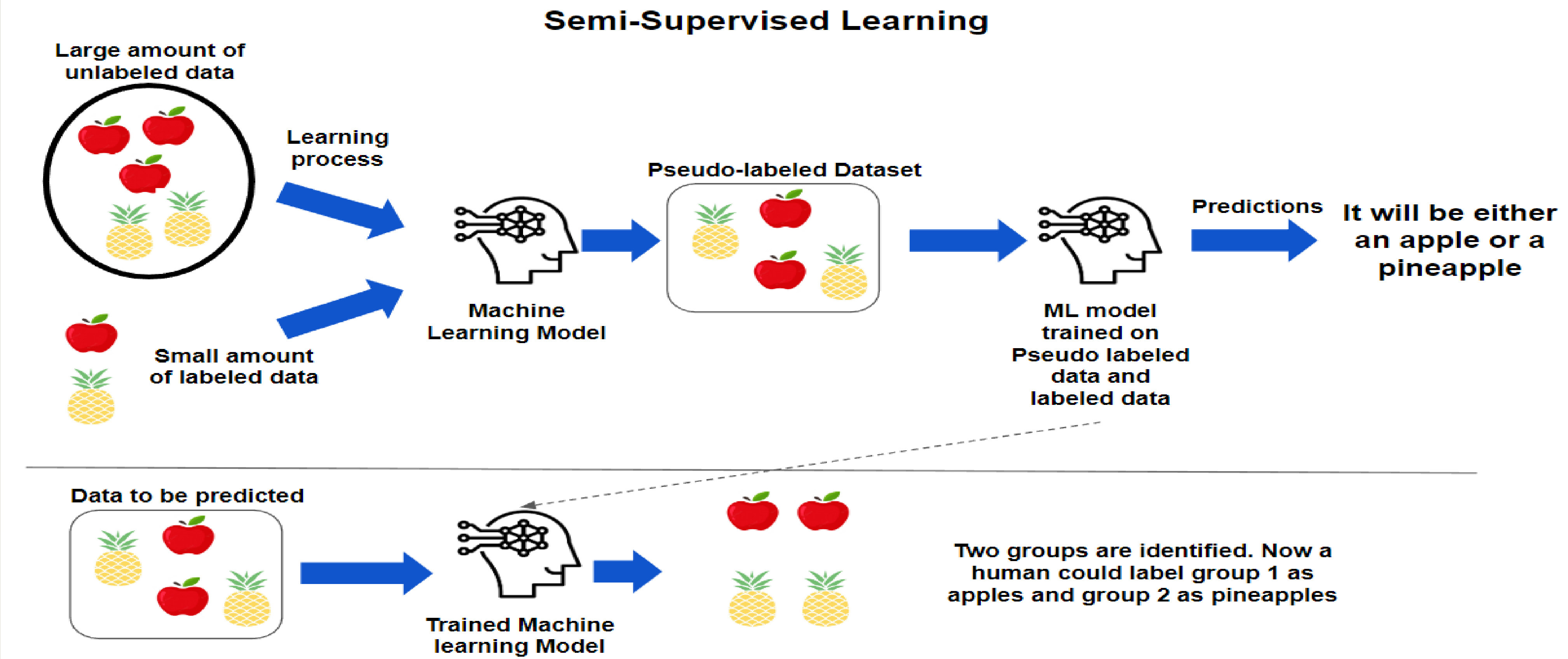


REINFORCEMENT LEARNING





SEMI-SUPERVISED LEARNING



WEEK 4, 5
DATASET OF MACHINE LEARNING, IMPOTENCE, HOW DATA ORGANIZED FOR ML ALGORITHM???,
PROPERTIES OF DATASET

DATASET OF MACHINE LEARNING

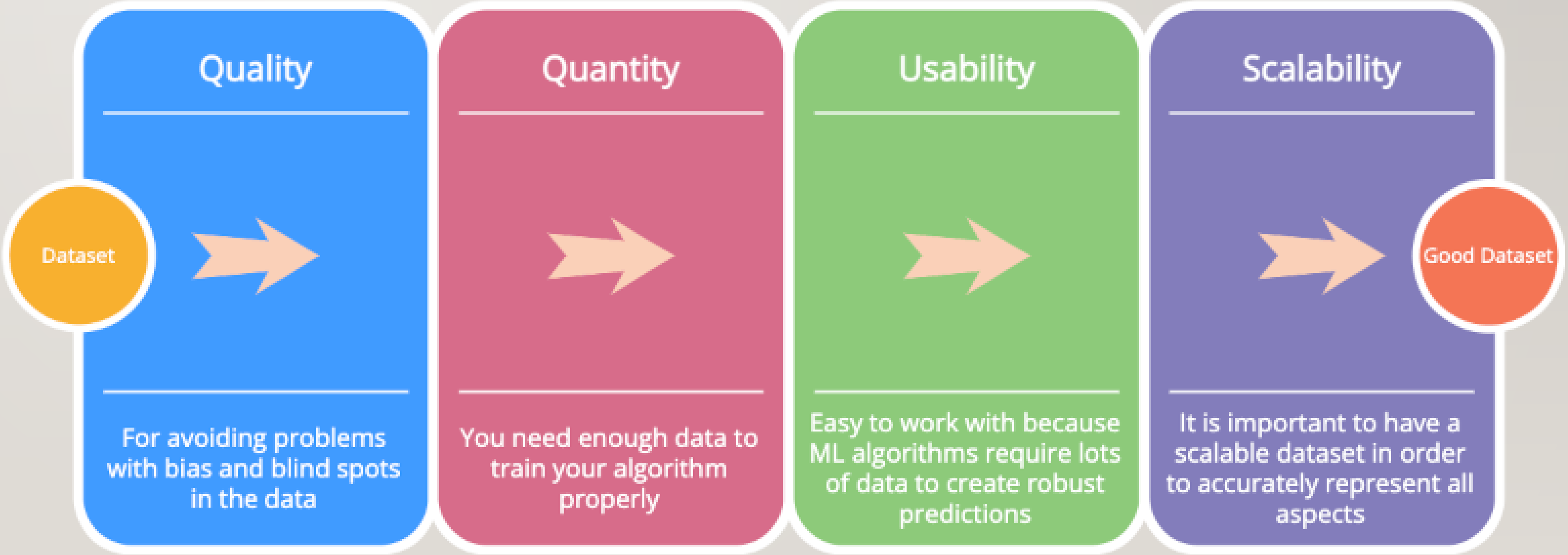
- A machine learning dataset is, quite simply, a collection of data pieces that can be treated by a computer as a single unit for analytic and prediction purposes. This means that the data collected should be made uniform and understandable for a machine that doesn’t see data the same way as humans do.
- After collecting the data, it’s important to preprocess it by cleaning and completing it, as well as annotate the data by adding meaningful tags readable by a computer.

WHAT IS DATASET ANALYSIS IN ML?

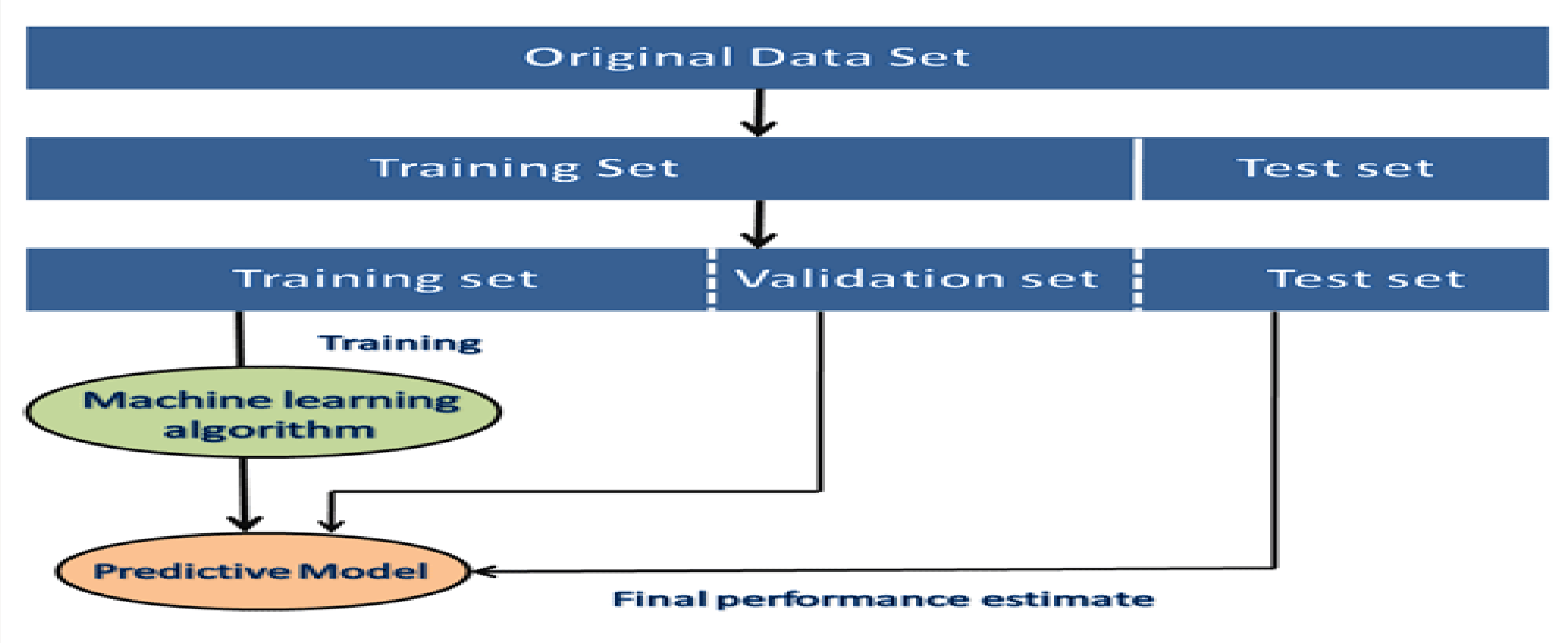
- Dataset analysis in machine learning involves inspecting, cleansing, transforming, and modeling data with the aim of discovering useful information by informing conclusions and supporting decision-making. A dataset is analyzed for further training of an ML algorithm.

WHY IS THE DATASET IMPORTANT IN ML?

- The dataset lays the groundwork for machine learning, shaping the model's ability to learn and make accurate predictions. First, a dataset allows you to train your ML model, and, second, it provides a benchmark for measuring the accuracy of the model.



HOW DATA ORGANIZED FOR ML ALGORITHM???



HOW DATA ORGANIZED FOR ML ALGORITHM???

Feature

Instance

x	y	z	class
0.5351795492	0.9443102776	0.1582435145	1
0.2372136153	0.6406416746	0.2375491596	1
0.9115356348	0.3311024322	0.5615073269	0
0.5634070287	0.4183148035	0.151904445	0
0.3728975195	0.3816657621	0.616341473	1
0.6783527289	0.938524515	0.5269012505	1
0.09568660734	0.04465749689	0.0133451798	0
0.2173318229	0.6170559076	0.3122273853	1
0.818890594	0.7459451367	0.9026713492	0
0.6064854042	0.5945985792	0.2188024961	0
0.1546966824	0.1579937453	0.1333579164	0

Train Dataset

Test Dataset

Id	SepalLeng	SepalWid	PetalLeng	PetalWidt	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa

TYPES OF DATASETS

- **Numerical Dataset:** They include numerical data points that can be solved with equations. These include temperature, humidity, marks and so on.
- **Categorical Dataset:** These include categories such as colour, gender, occupation, games, sports and so on.
- **Web Dataset:** These include datasets created by calling APIs using HTTP requests and populating them with values for data analysis. These are mostly stored in JSON (JavaScript Object Notation) formats.
- **Time series Dataset:** These include datasets between a period, for example, changes in geographical terrain over time.
- **Image Dataset:** It includes a dataset consisting of images. This is mostly used to differentiate the types of diseases, heart conditions and so on.

TYPES OF DATASETS

- **Ordered Dataset:** These datasets contain data that are ordered in ranks, for example, customer reviews, movie ratings and so on.
- **Partitioned Dataset:** These datasets have data points segregated into different members or different partitions.
- **File-Based Datasets:** These datasets are stored in files, in Excel as .csv, or .xlsx files.
- **Bivariate Dataset:** In this dataset, 2 classes or features are directly correlated to each other. For example, height and weight in a dataset are directly related to each other.
- **Multivariate Dataset:** In these types of datasets, as the name suggests 2 or more classes are directly correlated to each other. For example, attendance, and assignment grades are directly correlated to a student's overall grade.

PROPERTIES OF DATASET

- **Center of data:** This refers to the “middle” value of the data, often measured by mean, median, or mode. It helps understand where most of the data points are concentrated.

- **Skewness of data:** This indicates how symmetrical the data distribution is. A perfectly symmetrical distribution (like a normal distribution) has a skewness of 0. Positive skewness means the data is clustered towards the left, while negative skewness means it’s clustered towards the right.

- **Spread among data members:** This describes how much the data points vary from the center. Common measures include standard deviation or variance, which quantify how far individual points deviate from the average.

PROPERTIES OF DATASET

- **Presence of outliers:** These are data points that fall significantly outside the overall pattern. Identifying outliers can be important as they might influence analysis results and require further investigation.

- **Correlation among the data:** This refers to the strength and direction of relationships between different variables in the dataset. A positive correlation indicates values in one variable tend to increase as the other does, while a negative correlation suggests they move in opposite directions. No correlation means there's no linear relationship between the variables.

- **Type of probability distribution that the data follows:** Understanding the distribution (e.g., normal, uniform, binomial) helps us predict how likely it is to find certain values within the data and choose appropriate statistical methods for analysis.

WEEK 7, 8

K-NEAREST NEIGHBOR(KNN) ALGORITHM, INTUITION BEHIND KNN ALGORITHM, DISTANCE METRICS USED IN KNN ALGORITHM

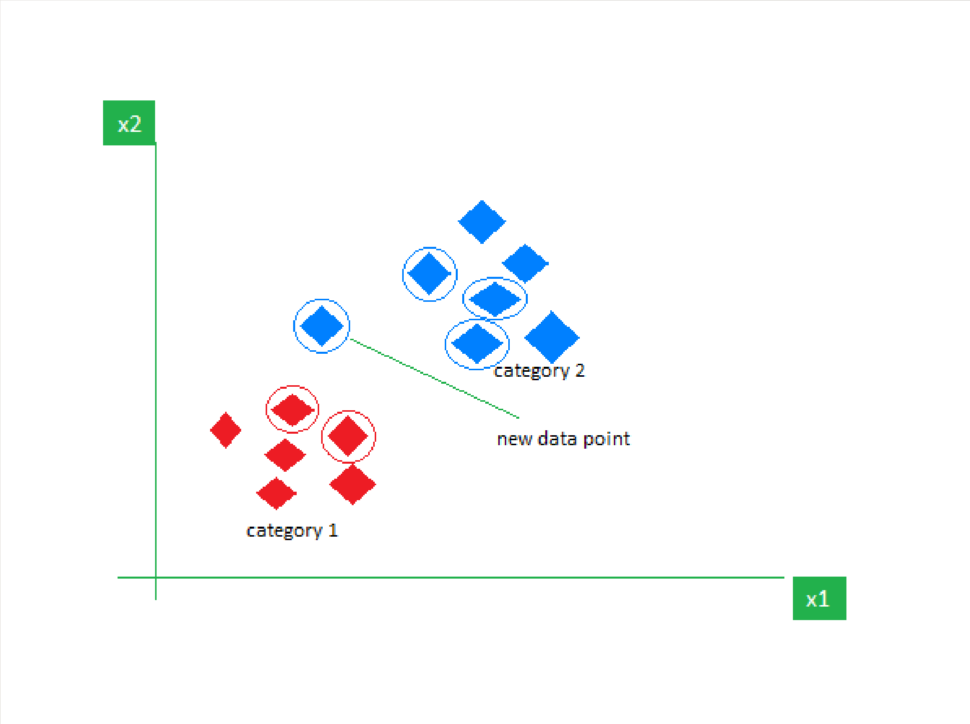
K-NEAREST NEIGHBOR(KNN) ALGORITHM

- The **K-Nearest Neighbors (KNN) algorithm** is a supervised machine learning method employed to tackle classification and regression problems.
- KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.

INTUITION BEHIND KNN ALGORITHM

- If we plot these points on a graph, we may be able to locate some clusters or groups. Now, given an unclassified point, we can assign it to a group by observing what group its nearest neighbors belong to. This means a point close to a cluster of points classified as ‘Red’ has a higher probability of getting classified as ‘Red’.

- Intuitively, we can see that the first point (2.5, 7) should be classified as ‘Green’, and the second point (5.5, 4.5) should be classified as ‘Red’.

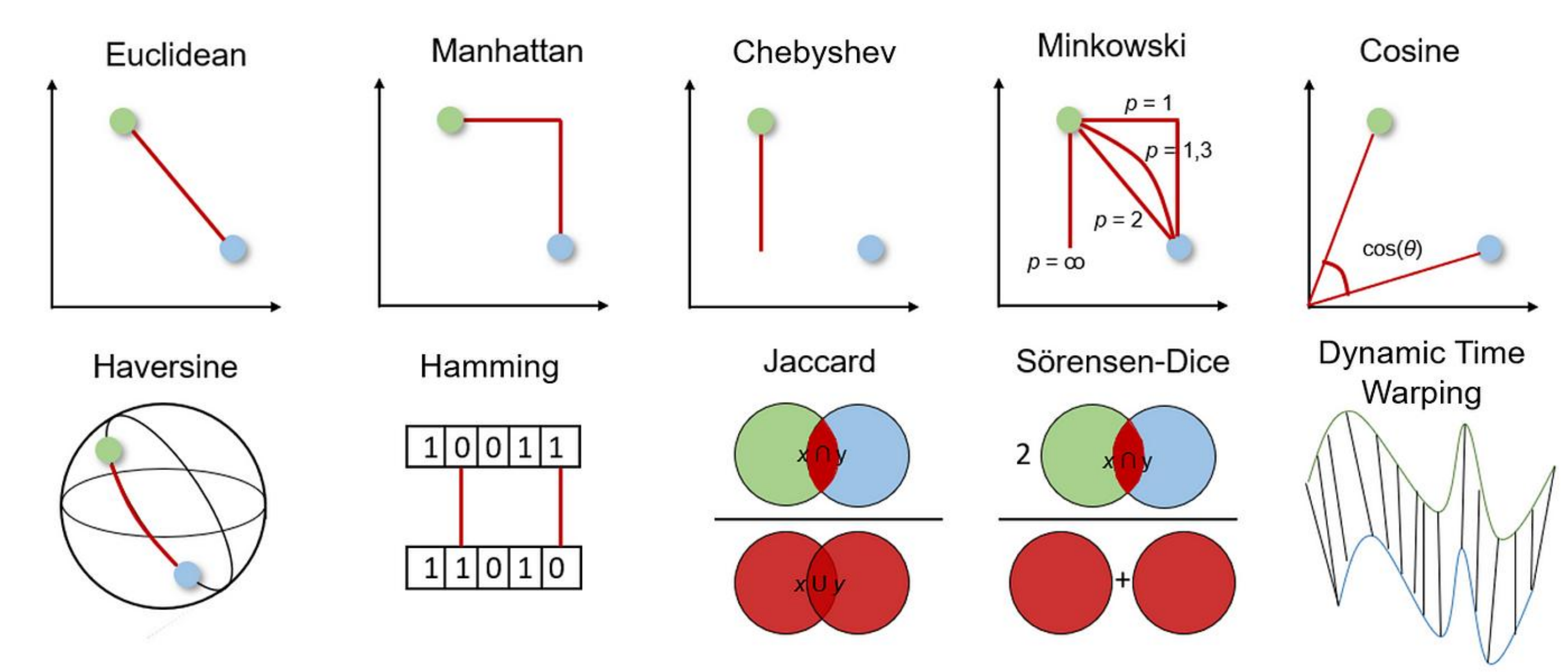


WHY DO WE NEED A KNN ALGORITHM?

- (K-NN) algorithm is a versatile and widely used machine learning algorithm that is primarily used for its simplicity and ease of implementation. It does not require any assumptions about the underlying data distribution. It can also handle both numerical and categorical data, making it a flexible choice for various types of datasets in classification and regression tasks.
- The K-NN algorithm works by finding the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance. The class or value of the data point is then determined by the majority vote or average of the K neighbors. This approach allows the algorithm to adapt to different patterns and make predictions based on the local structure of the data.

DISTANCE METRICS USED IN KNN ALGORITHM

- ❖Euclidean Distance
- ❖Manhattan Distance
- ❖Minkowski Distance



Nearest-neighbor algorithm

a) A pseudo code for the nearest neighbor algorithm is

ALGORITHM *Nearest-neighbor* $(D[1...n,1...n], s)$

//Input: A $n \times n$ distance matrix $D[1...n,1...n]$ and an index s of the starting city.

//Output: A list Path of the vertices containing the tour is obtained.

for $i \leftarrow 1$ to n do Visited $[i] \leftarrow$ false

Initialize the list Path with s

Visited $[s] \leftarrow$ true

Current $\leftarrow s$

for $i \leftarrow 2$ to n do

    Find the lowest element in row current and unmarked column j containing the element.

    Current $\leftarrow j$

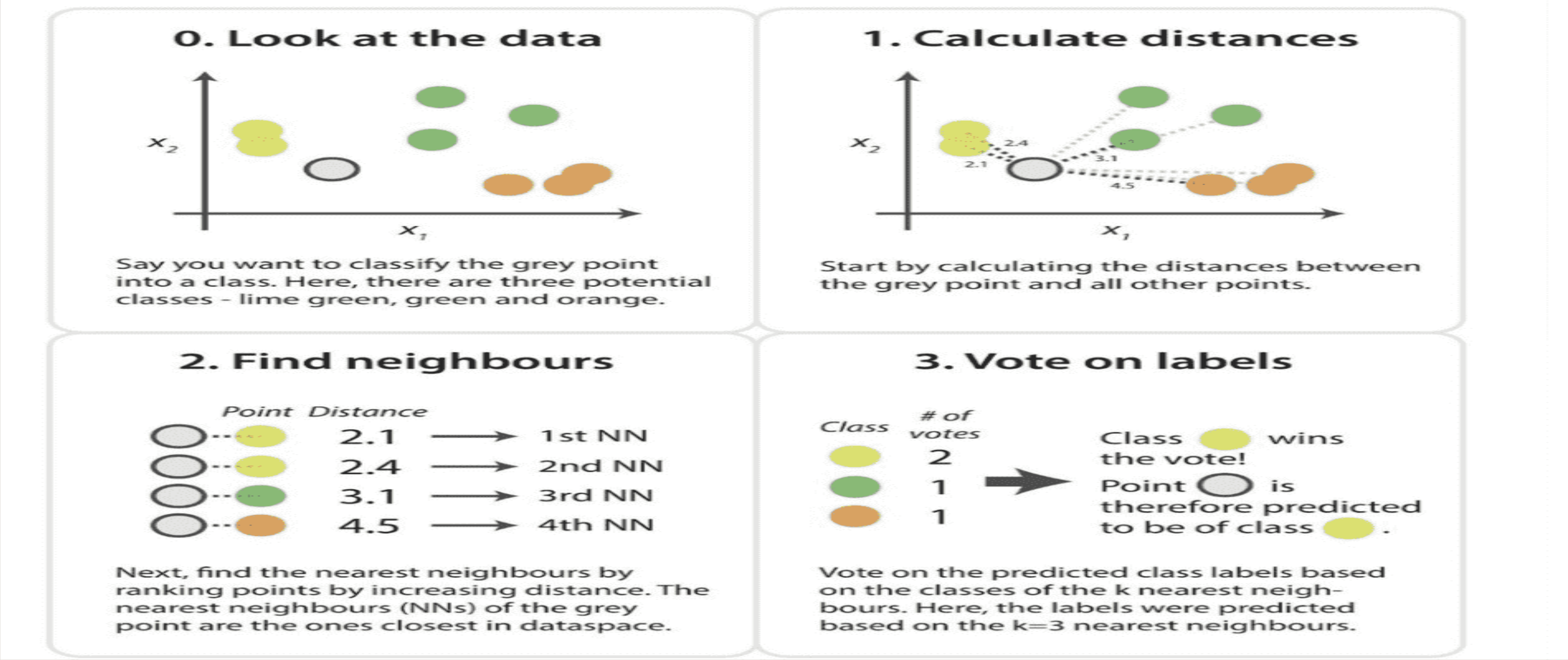
    Visited $[j] \leftarrow$ true

    Add j to the end of list Path

Add s to the end of list Path

return Path

KNN ALGORITHM PROCEDURE



How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors.
Step-2: Calculate the Euclidean distance of K number of neighbors.
Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
Step-4: Among these k neighbors, count the number of the data points in each category.
Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
Step-6: Our model is ready.

KNN MATH EXAMPLE

Example:

Height(cm)	Weight(KG)	Class	Distance	Rank
169	58	Normal	1.4	1
170	55	Normal	2	2
173	57	Normal	3	3
174	56	Underweight	4.1	4
167	51	Underweight	6.7	5
173	64	Normal	7.6	6
170	57	?		

KNN MATH EXAMPLE

The Distance Formula:

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

$$d1=\sqrt{(170-169)^2+(57-51)^2}=6.8$$
$$d2=\sqrt{(170-170)^2+(57-55)^2}=2$$
$$d3=\sqrt{(170-173)^2+(57-57)^2}=3$$
$$d4=\sqrt{(170-174)^2+(57-57)^2}=4$$

now lets calculate the nearest neighbor at K= 3

170 cm 57 kg = Normal

ADVANTAGES & DISADVANTAGES OF KNN

Advantages of KNN Algorithm:

1. It is simple to implement.
2. It is robust to the noisy training data
3. It can be more effective if the training data is large.

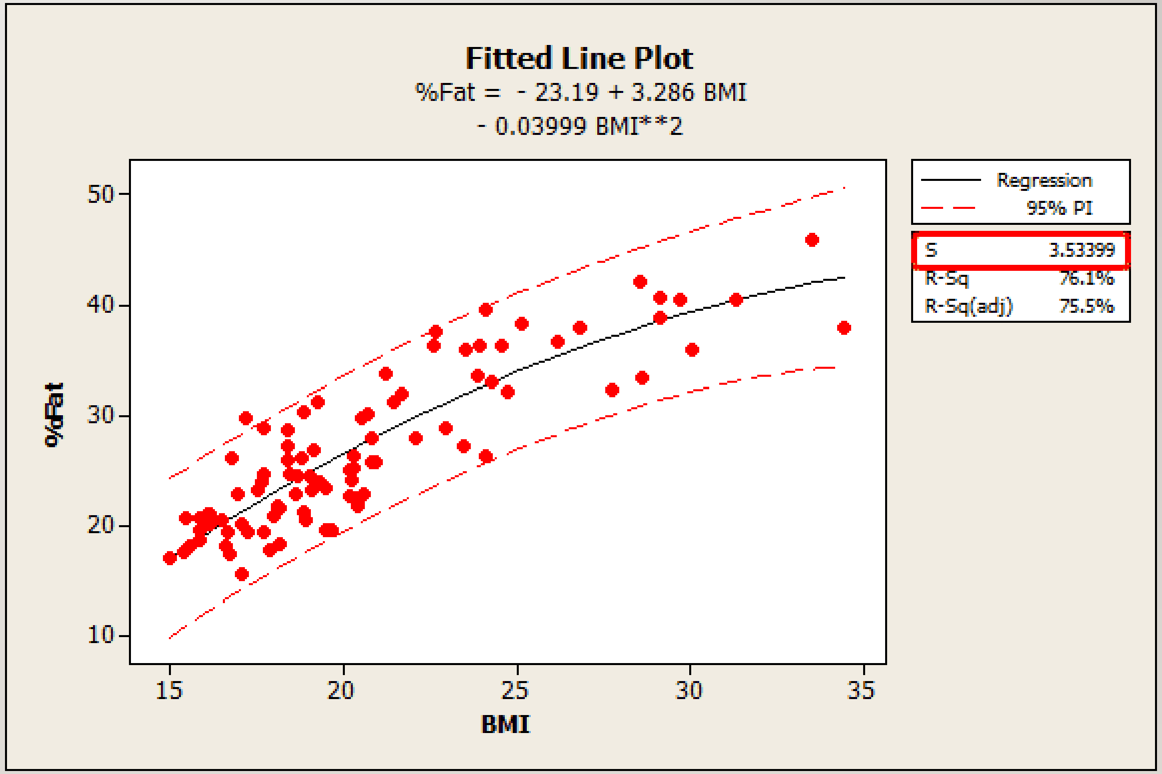
Disadvantages of KNN Algorithm:

1. Always needs to determine the value of K which may be complex some time.
2. The computation cost is high because of calculating the distance between the data points for all the training samples.

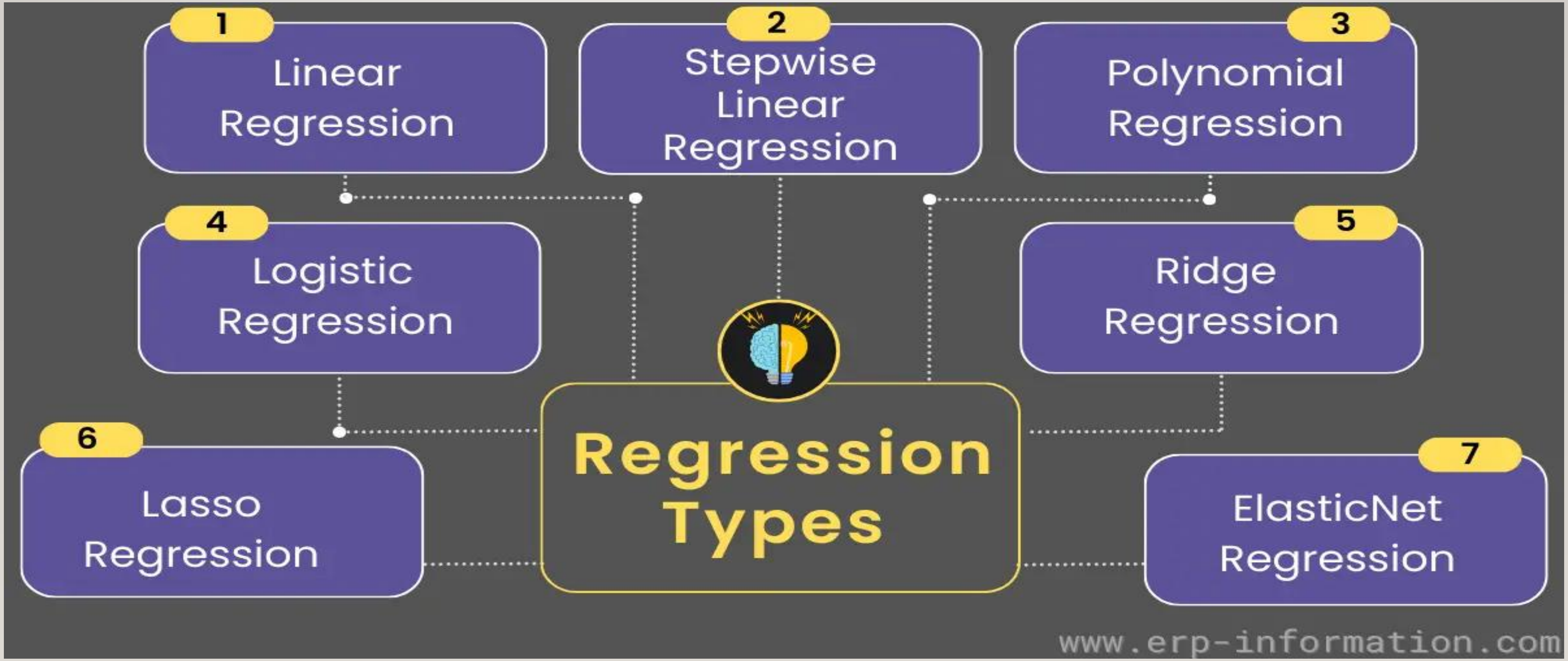
WEEK 9,10
REGRESSION ANALYSIS, TYPE REGRESSION ANALYSIS, LINEAR REGRESSION, THE LEAST SQUARES (REGRESSION)

REGRESSION ANALYSIS

- **What Is Regression?**
  - Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between a dependent variable and one or more independent variables.



TYPE REGRESSION ANALYSIS



LINEAR REGRESSION

- Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data.
- When there is only one independent feature, it is known as Simple Linear Regression, and when there are more than one feature, it is known as Multiple Linear Regression.

THE MODEL

The model has a deterministic and a probabilistic components

House
Cost

Building a house costs about
$75 per square foot.
House cost = 25000 + 75(Size)

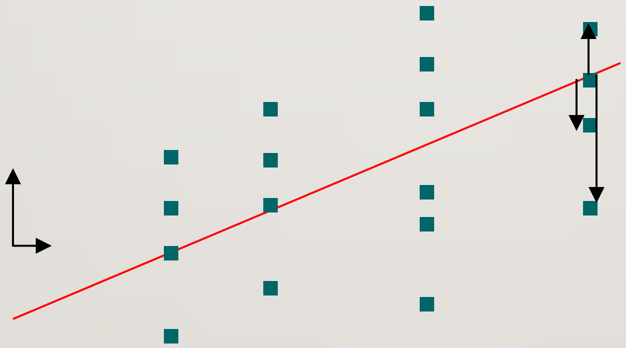
Most lots sell
for $25,000

House size

However, house cost vary even among same size houses!

Since cost behave unpredictably, we add a random component.

House Cost



Most lots sell for $25,000

$$\text{House cost} = 25000 + 75\,(\text{Size}) + \varepsilon$$

House size

- The first order linear model

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Y = dependent variable

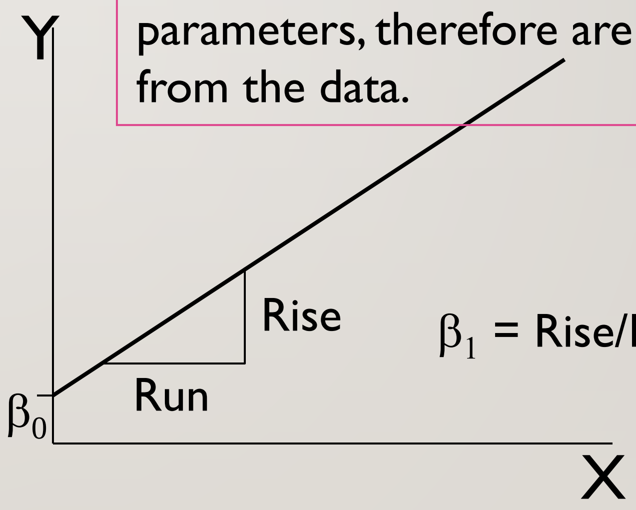
X = independent variable

β_0 = Y-intercept

β_1 = slope of the line

ε = error variable

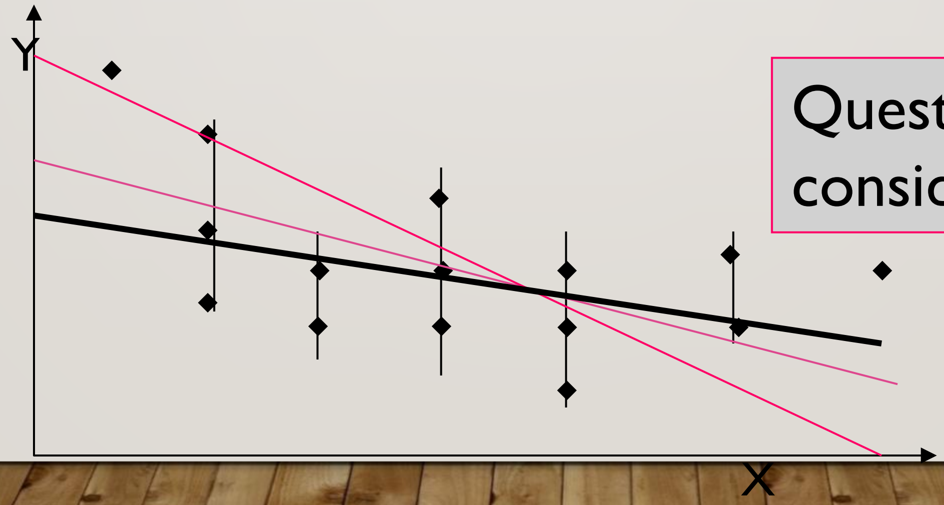
β_0 and β_1 are unknown population parameters, therefore are estimated from the data.



β_1 = Rise/Run

ESTIMATING THE COEFFICIENTS

- The estimates are determined by
  - drawing a sample from the population of interest,
  - calculating sample statistics.
  - producing a straight line that cuts into the data.



Question: What should be considered a good line?

THE LEAST SQUARES (REGRESSION) LINE

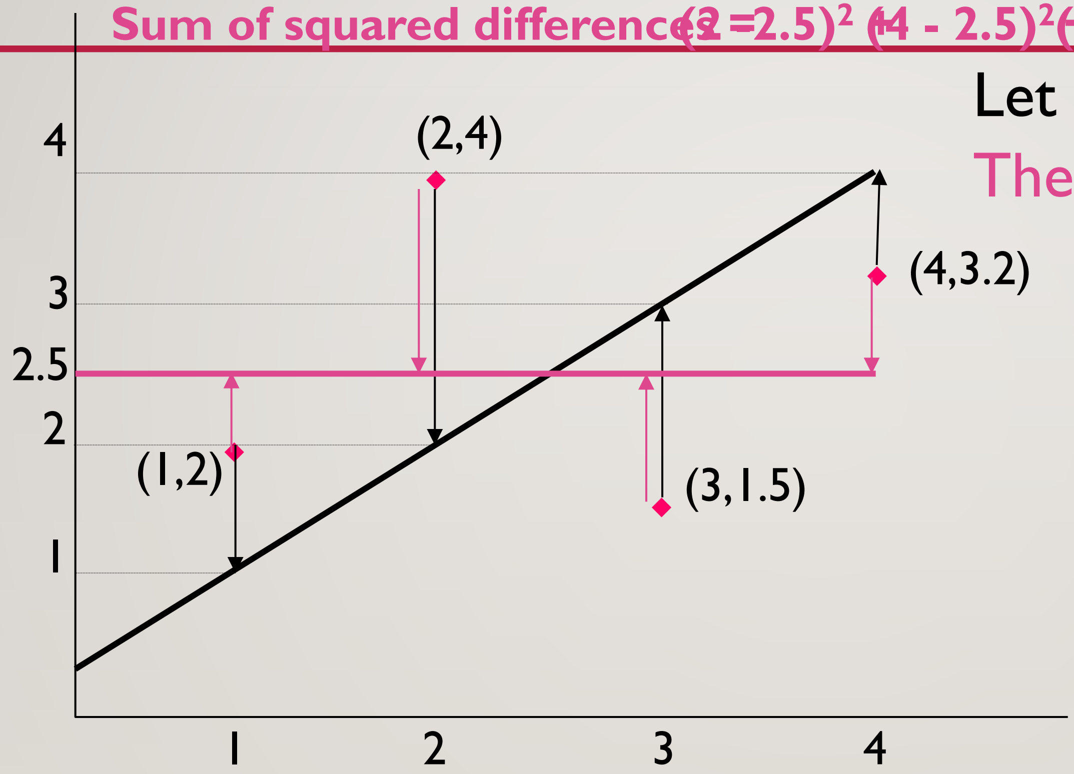
A good line is one that minimizes
the sum of squared differences between the points and the line.

Sum of squared differences = $(2 - 1)^2 + (4 - 2)^2 + (1.5 - 3)^2 + (3.2 - 4)^2 = 6.89$

Sum of squared differences = $(2 - 2.5)^2 + (4 - 2.5)^2 + (1.5 - 2.5)^2 + (3.2 - 2.5)^2 = 3.99$

Let us compare two lines

The second line is horizontal



The smaller the sum of squared differences the better the fit of the line to the data.

THE ESTIMATED COEFFICIENTS

To calculate the estimates of the line coefficients, that minimize the differences between the data points and the line, use the formulas:

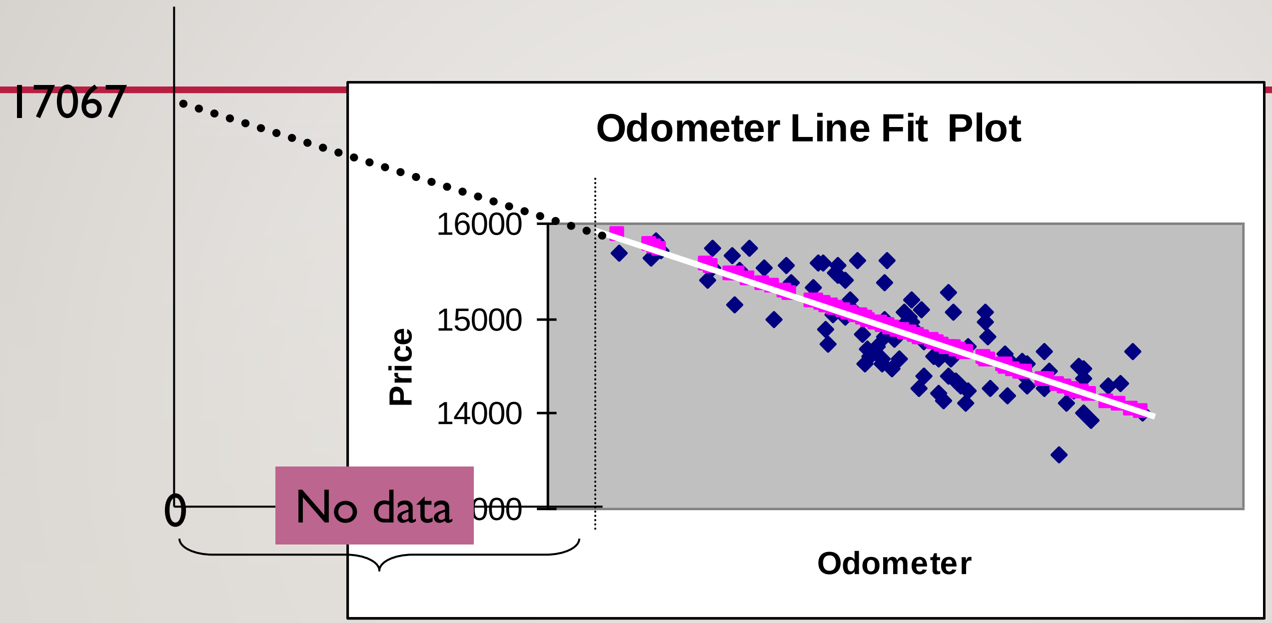
$$b_1 = \frac{\mathrm{cov}(X,Y)}{s_X^2}\left(= \frac{s_{XY}}{s_X^2}\right)$$

$$b_0 = \bar{Y} - b_1\bar{X}$$

The regression equation that estimates the equation of the first order linear model is:

$$\hat{Y} = b_0 + b_1 X$$

INTERPRETING THE LINEAR REGRESSION -EQUATION



$$\hat{Y} = 17{,}067 - .0623\,X$$

The intercept is b_0 = $17067.

Do not interpret the intercept as the "Price of cars that have not been driven"

This is the slope of the line.
For each additional mile on the odometer, the price decreases by an average of $0.0623

MATH EQUATION DERIVATION OF LINEAR REGRESSION

- **For Math equation please see :**
  - https://www.youtube.com/watch?v=DSQ2plMtbLc&t=431s
- **For Example :**
  - https://www.geeksforgeeks.org/linear-regression-formula/

WEEK 11, 12
CONFUSION MATRIX, ACCURACY, PRECISION, RECALL, F1 SCORE

What is a confusion matrix?

It is a tool to evaluate performance of supervised machine learning algorithm when used for classification problem.

Consider....

You train a ML model for a data set that is labelled only for YES or NO for each data row.

Ex. Loan worthiness of a person against their income, education, credit score, marital status, number of children and other features.

But remember....

Only for this problem our classes are YES and NO. In general classes can be any two different things such as Positive or Negative, Bus or Truck, Spam or Not Spam, Dog or Cat and so on.

NOTE: Classes do not have to be necessarily opposite to each other.

Now how do you check the performance of your model on test data?

YES!

With the help of confusion matrix.

Let us assume your test data contained 100 rows. Then your confusion matrix might look something like this...

		Predicted: Yes	Predicted: No
Actual	Yes	45	15
	No	10	30

$n = 100$

Oh before that let us consider ....

YES to be positive label

No to be negative label

True Positives

		Predicted: Yes	Predicted: No
Actual	Yes	45	15
	No	10	30

$n = 100$

TP ✓

When model predicts YES and actual value is also YES

		Predicted Yes	Predicted No
Actual	Yes	45	15
	No	10	30

$n = 100$

FP !

False Positives or Type 1 error

When model predicts YES but actual value is NO

		Predicted Yes	Predicted No
Actual	Yes	45	15
	No	10	30

$n = 100$

TN ✓

True Negatives

When model predicts NO and actual value is also NO

		Predicted	
		Yes	No
Actual	Yes	45	15
	No	10	30

$n = 100$

FN !

False Negatives or Type II error

When model predicts NO but actual value is YES

Need for Confusion Matrix

- Evaluates the performance of the classification models, making predictions on test data, and tells how good classification model is.
- Tells the error made by the classifiers but also the type of errors such as either type-I or type-II error.
- Helpful to calculate the parameters for classification model, such as accuracy, precision, etc.

Sensitivity and Specificity

- Sensitivity tells us what proportion of the positive class got correctly classified.
- A simple example would be determining what proportion of the actual sick people were correctly detected by the model.
- False Negative Rate (FNR) tells us what proportion of the positive class got incorrectly classified by the classifier.
- A higher TPR and lower FNR are desirable since we want to classify the positive class correctly.
- Specificity tells us what proportion of the negative class got correctly classified.

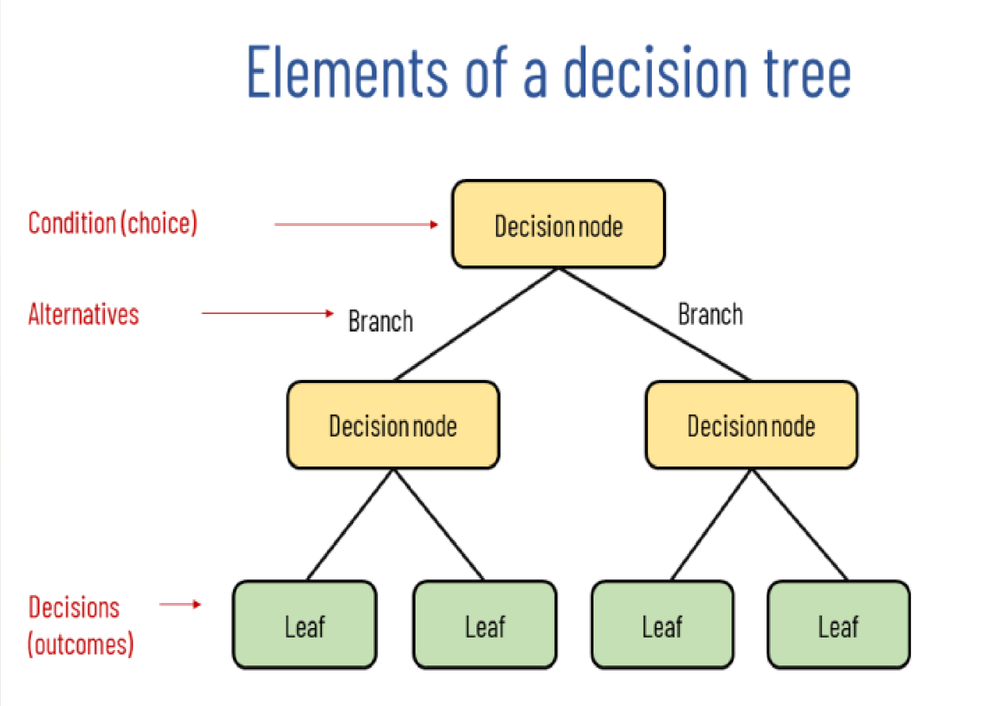
Calculation using Confusion Matrix

- Classification Accuracv: Defines how often the model predicts the correct output $\text{Accuracy}=\frac{TP+TN}{TP+FP+FN+TN}$
- Misclassification rate: Also termed as Error rate, and it defines how often the model gives the wrong predictions. $\text{Error rate}=\frac{FP+FN}{TP+FP+FN+TN}$
- Precision: It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model. $\text{Precision}=\frac{TP}{TP+FP}$
- Recall: It is defined as the out of total positive classes, how our model predicted correctly. $\text{Recall}=\frac{TP}{TP+FN}$
- F-measure: Score helps us to evaluate the recall and precision at the same time. $\text{F-measure}=\frac{2*Recall*Precision}{Recall+Precision}$

WEEK 13, 14
DECISION TREE ALGORITHM

DECISION TREE ALGORITHM

- **What is a Decision Tree?**
  - A **decision tree** is a flowchart-like structure used to make decisions or predictions. It consists of nodes representing decisions or tests on attributes, branches representing the outcome of these decisions, and leaf nodes representing final outcomes or predictions. Each internal node corresponds to a test on an attribute, each branch corresponds to the result of the test, and each leaf node corresponds to a class label or a continuous value.



STRUCTURE OF A DECISION TREE

1. **Root Node**: Represents the entire dataset and the initial decision to be made.
2. **Internal Nodes**: Represent decisions or tests on attributes. Each internal node has one or more branches.
3. **Branches**: Represent the outcome of a decision or test, leading to another node.
4. **Leaf Nodes**: Represent the final decision or prediction. No further splits occur at these nodes.

HOW DECISION TREES WORK?

- The process of creating a decision tree involves:

  1. **Selecting the Best Attribute**: Using a metric like Gini impurity, entropy, or information gain, the best attribute to split the data is selected.

  2. **Splitting the Dataset**: The dataset is split into subsets based on the selected attribute.

  3. **Repeating the Process**: The process is repeated recursively for each subset, creating a new internal node or leaf node until a stopping criterion is met (e.g., all instances in a node belong to the same class or a predefined depth is reached).

METRICS FOR SPLITTING

- **Gini Impurity**: Measures the likelihood of an incorrect classification of a new instance if it was randomly classified according to the distribution of classes in the dataset.
- **Entropy**: Measures the amount of uncertainty or impurity in the dataset.
- **Information Gain**: Measures the reduction in entropy or Gini impurity after a dataset is split on an attribute.

ID3 ALGORITHM

- ID3 stands for Iterative Dichotomiser 3 and is named such because the algorithm iteratively (repeatedly) dichotomizes(divides) features into two or more groups at each step.
- ID3 uses a **top-down greedy** approach to build a decision tree. In simple words, the **top-down** approach means that we start building the tree from the top and the **greedy** approach means that at each iteration we select the best feature at the present moment to create a node.

Entropy and Information gain

- The entropy is a measure of the randomness in the information being processed.
- If the sample is completely homogeneous the entropy is zero and if the sample is equally divided then it has entropy of one.
- Entropy can be calculated as:

$$Entropy(S) = \sum - p(I) \,.\, \log_2 p(I)$$

- The information gain is based on the decrease in entropy after a data-set is split on an attribute.
- Information gain can be calculated as:

$$Gain(S, A) = Entropy(S) - \sum [\, p(S|A) \,.\, Entropy(S|A) \,]$$

Decision tree for deciding if tennis is playable, using data from past 14 days

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

Entropy

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

$$\text{Entropy(Decision)} = -\,p(Yes)\,.\,\log_2 p(Yes) - p(No)\,.\,\log_2 p(No)$$

$$\text{Entropy(Decision)} = -\,(9/14)\,.\,\log_2(9/14) - (5/14)\,.\,\log_2(5/14)$$

$$= \mathbf{0.940}$$

Wind factor on decision

- Wind attribute has two labels: **weak** and **strong**.

Gain(Decision, Wind) = Entropy(Decision) – [p(Decision|Wind=Weak) . Entropy(Decision|Wind=Weak)] – [p(Decision|Wind=Strong) . Entropy(Decision|Wind=Strong)]

- We need to calculate **(Decision|Wind=Weak)** and **(Decision|Wind=Strong)** respectively.

Weak wind factor

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
13	Overcast	Hot	Normal	Weak	Yes

- There are 8 instances for weak wind. Decision of 2 items are **no** and 6 items are **yes**.
- Entropy(Decision|Wind=Weak) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)

Entropy(Decision|Wind=Weak) = – (2/8) . $\log_2$(2/8) – (6/8) . $\log_2$(6/8)

=**0.811**

Strong wind factor

Day	Outlook	Temp.	Humidity	Wind	Decision
2	Sunny	Hot	High	Strong	No
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
14	Rain	Mild	High	Strong	No

- Here, there are 6 instances for strong wind. Decision is divided into two equal parts.

$$\text{Entropy}(\text{Decision}|\text{Wind=Strong}) = -(3/6)\ .\ \log_2(3/6) - (3/6).\log_2(3/6)$$

$$= 1$$

Wind factor on decision

- Information Gain can be calculated as:

Gain(Decision, Wind) = Entropy(Decision) – [p(Decision|Wind=Weak) . Entropy(Decision|Wind=Weak)] – [p(Decision|Wind=Strong) . Entropy(Decision|Wind=Strong)]

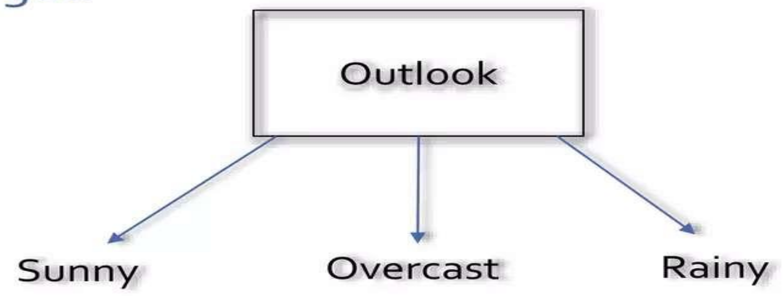
= 0.940 – [(8/14) . 0.811] – [(6/14). 1]

= **0.048**

Other factor on decision

On applying similar calculation on the other columns, we get:

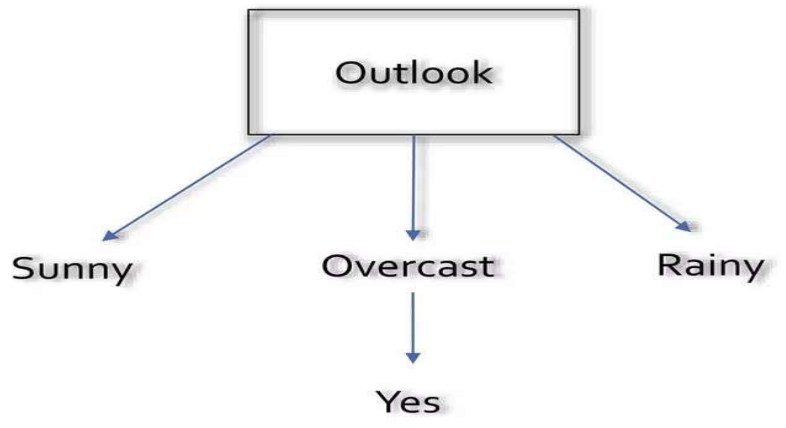
- Gain(Decision, Outlook) = 0.246
- Gain(Decision, Temperature) = 0.029
- Gain(Decision, Humidity) = 0.151



Overcast outlook on decision

- Decision will always be yes if outlook were overcast.

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

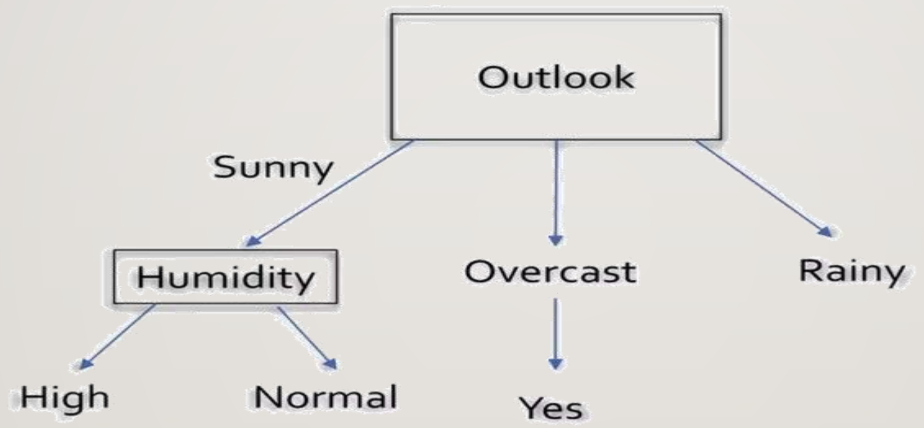

Outlook
Sunny
Overcast
Rainy
Yes

Sunny outlook on decision

We have 5 instances for sunny outlook. Decision would be probably **3/5 percent no, 2/5 percent yes**

- Gain(Outlook=Sunny|Temperature) = **0.570**
- Gain(Outlook=Sunny|Humidity) = **0.970**
- Gain(Outlook=Sunny|Wind) = **0.019**

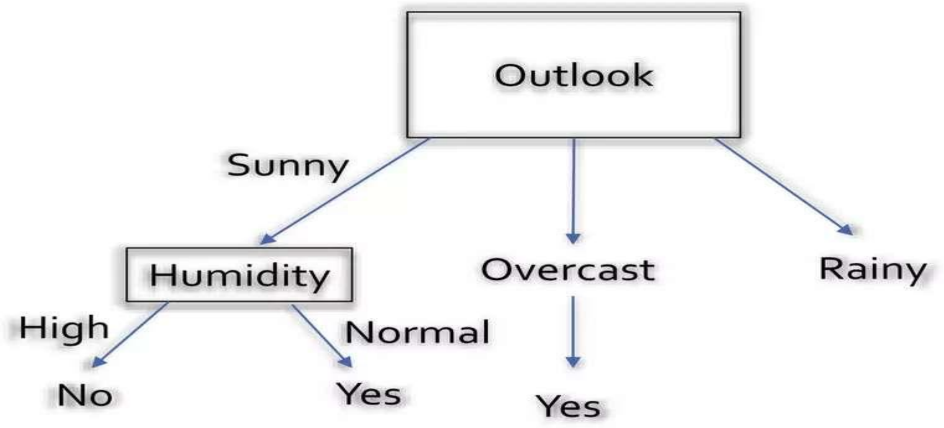
Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes


Outlook
Sunny
Humidity
Overcast
Rainy
High
Normal
Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

- Decision will always be **no** when humidity is **high**.
- Decision will always be **yes** when humidity is **normal**.

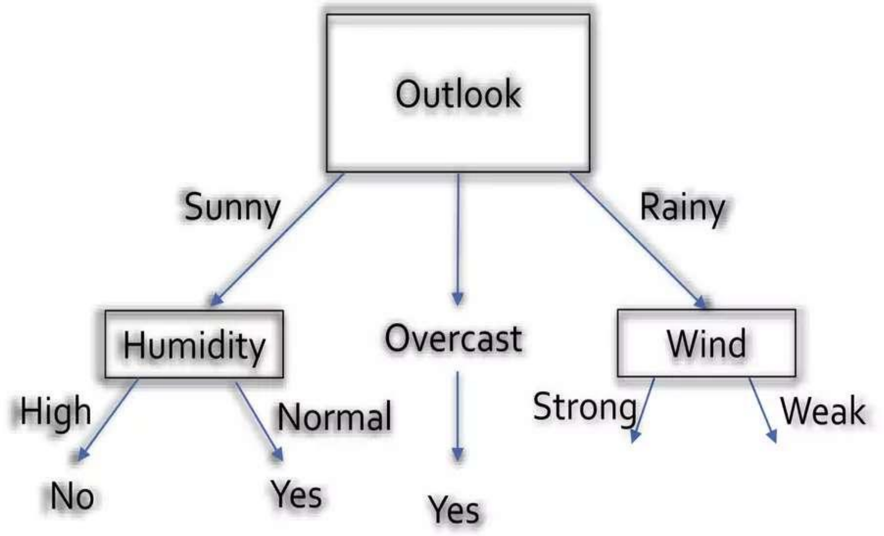

Outlook
Sunny
Humidity
Overcast
Rainy
High
Normal
No
Yes
Yes

Rain outlook on decision

Information gain for Rain outlook are:

- Gain(Outlook=Rain | Temperature) = 0.02
- Gain(Outlook=Rain | Humidity) = 0.02
- Gain(Outlook=Rain | Wind) = 0.971

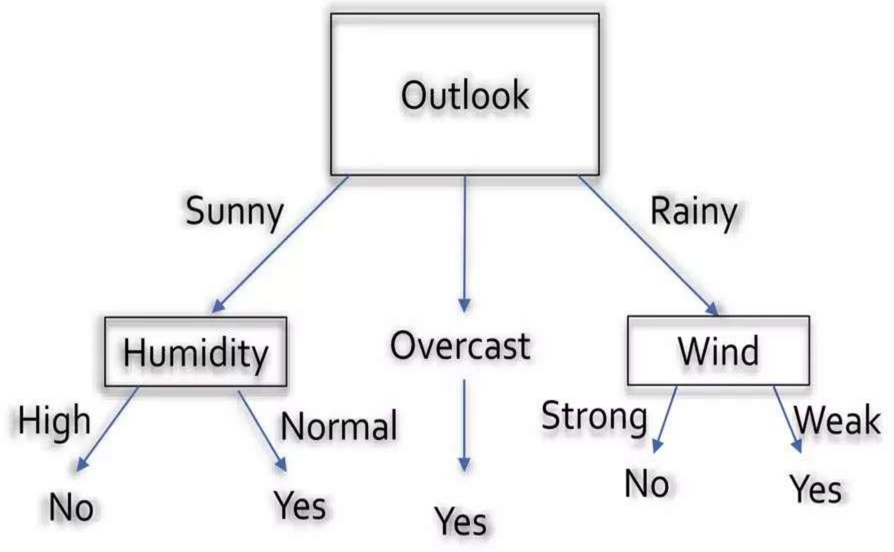
Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No


Outlook
Sunny
Rainy
Humidity
Overcast
Wind
High
Normal
Strong
Weak
No
Yes
Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

- Decision will always be **yes** if wind were **weak** and outlook were **rain**.
- Decision will always be **no** if wind were **strong** and outlook were **rain**.


Outlook
Sunny
Rainy
Humidity
Overcast
Wind
High
Normal
Strong
Weak
No
Yes
Yes
No
Yes

EXERCISE…

Attribute				Classes
Gender	**Car Ownership**	**Travel Cost**	**Income Level**	**Transportation**
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

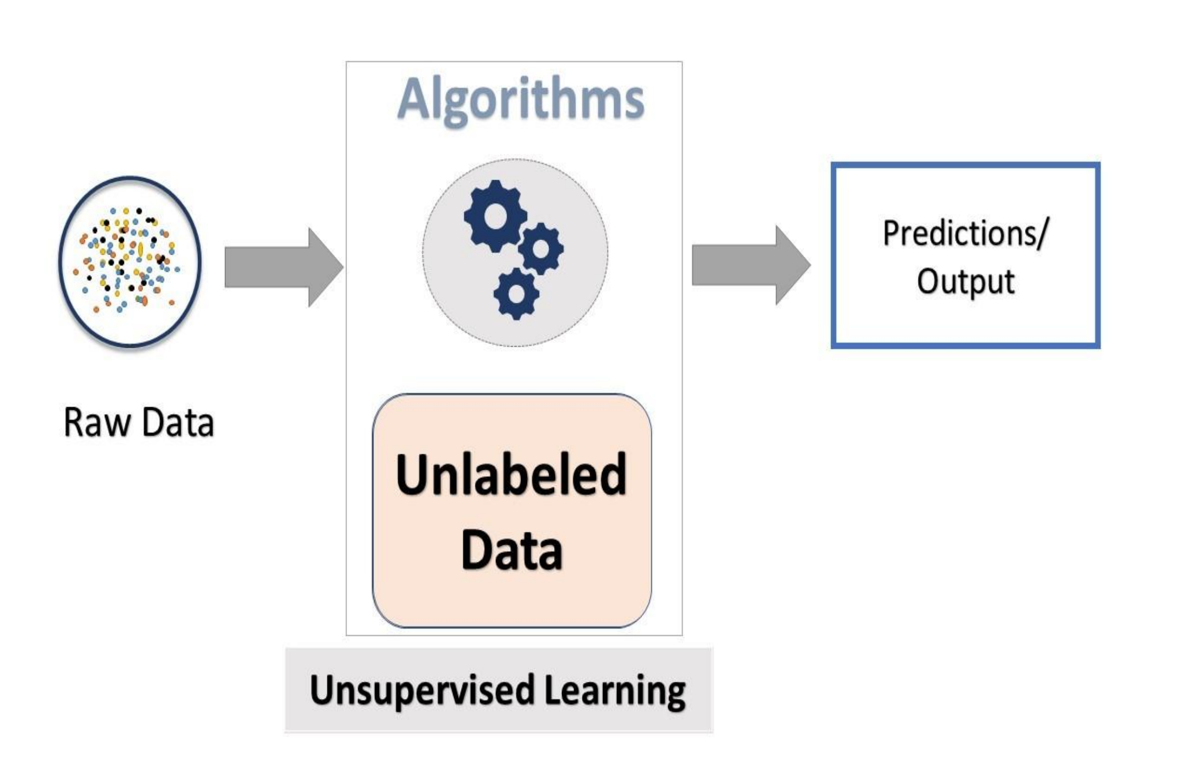
GIVEN TRAINING SET

We use non-category attributes to predict the values of category attributes.

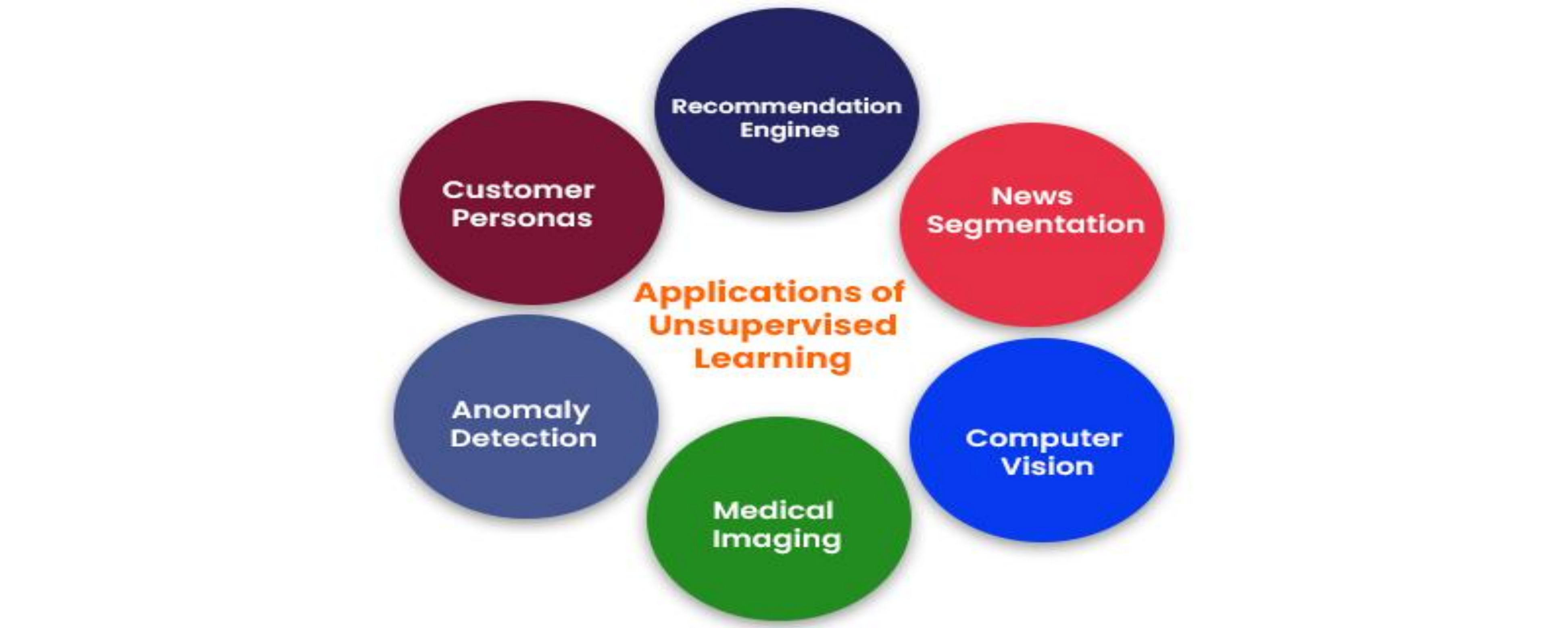
WEEK 14, 15
UNSUPERVISED LEARNING, CLUSTERING, TYPE OF CLUSTERING , KMEANS CLUSTERING

UNSUPERVISED LEARNING

- **Unsupervised learning** in artificial intelligence is a type of machine learning that learns from data without human supervision. Unlike supervised learning, unsupervised machine learning models are given unlabeled data and allowed to discover patterns and insights without any explicit guidance or instruction.



UNSUPERVISED LEARNING APPLICATION AREA



TYPES OF UNSUPERVISED LEARNING

Unsupervised Learning Algorithms

Clustering

- K-Means
- Polynomial
- Hierarchical
- Fuzzy C-Means

Dimensionality Reduction

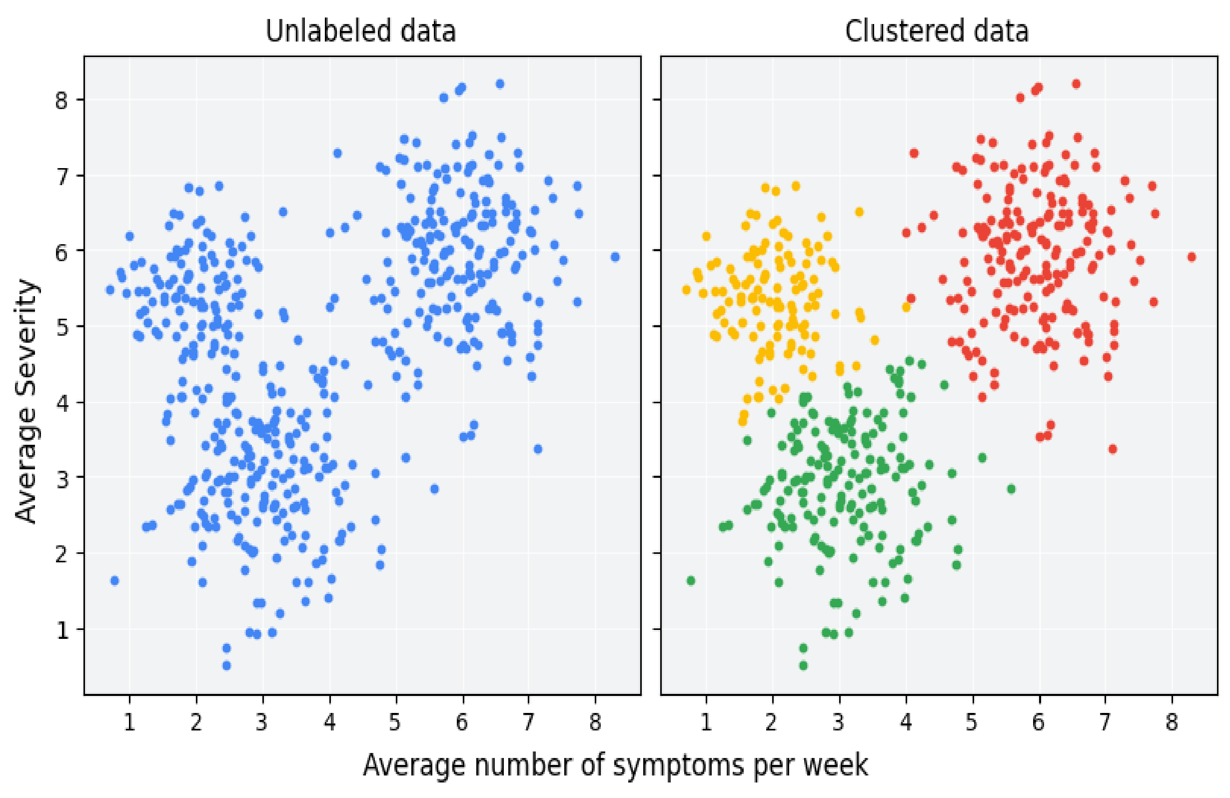
- Principal Component Analysis
- Kernel Principal Analysis

Association (Data Mining)

- Apriori Algorithm
- Eclat Algorithm
- FP-Growth Algorithm

CLUSTERING

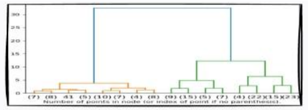
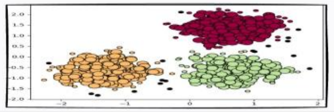
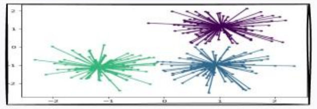
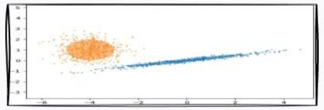
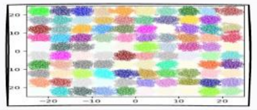
- Clustering is an unsupervised machine learning technique designed to group unlabeled examples based on their similarity to each other. (If the examples are labeled, this kind of grouping is called classification.) Consider a hypothetical patient study designed to evaluate a new treatment protocol.



TYPE OF CLUSTERING

6 Types of Clustering Algorithms in Machine Learning

blog.DailyDoseofDS.com

Clustering Algorithm Type		Clustering Methodology	Algorithm(s)
	Centroid-based	Cluster points based on proximity to centroid	KMeans KMeans++ KMedoids
	Connectivity-based	Cluster points based on proximity between clusters	Hierarchical Clustering (Agglomerative and Divisive)
	Density-based	Cluster points based on their density instead of proximity	DBSCAN OPTICS HDBSCAN
	Graph-based	Cluster points based on graph distance	Affinitiy Propagation Spectral Clustering
	Distribution-based	Cluster points based on their likelihood of belonging to the same distribution.	Gaussian Mixture Models (GMMs)
	Compression-based	Transform data to a lower dimensional space and then perform clustering	BIRCH

KMEANS CLUSTERING

- **K-means clustering** is a machine learning technique that groups data points into clusters based on their similarity. It's a centroid-based algorithm that uses a mathematical distance measure to assign data points to clusters. The goal is to minimize the distance between data points and their assigned clusters.

K-MEANS CLUSTERING

- The **k-means algorithm** is an algorithm to cluster n objects based on attributes into k partitions, where $k < n$.
- It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.
- It assumes that the object attributes form a vector space.

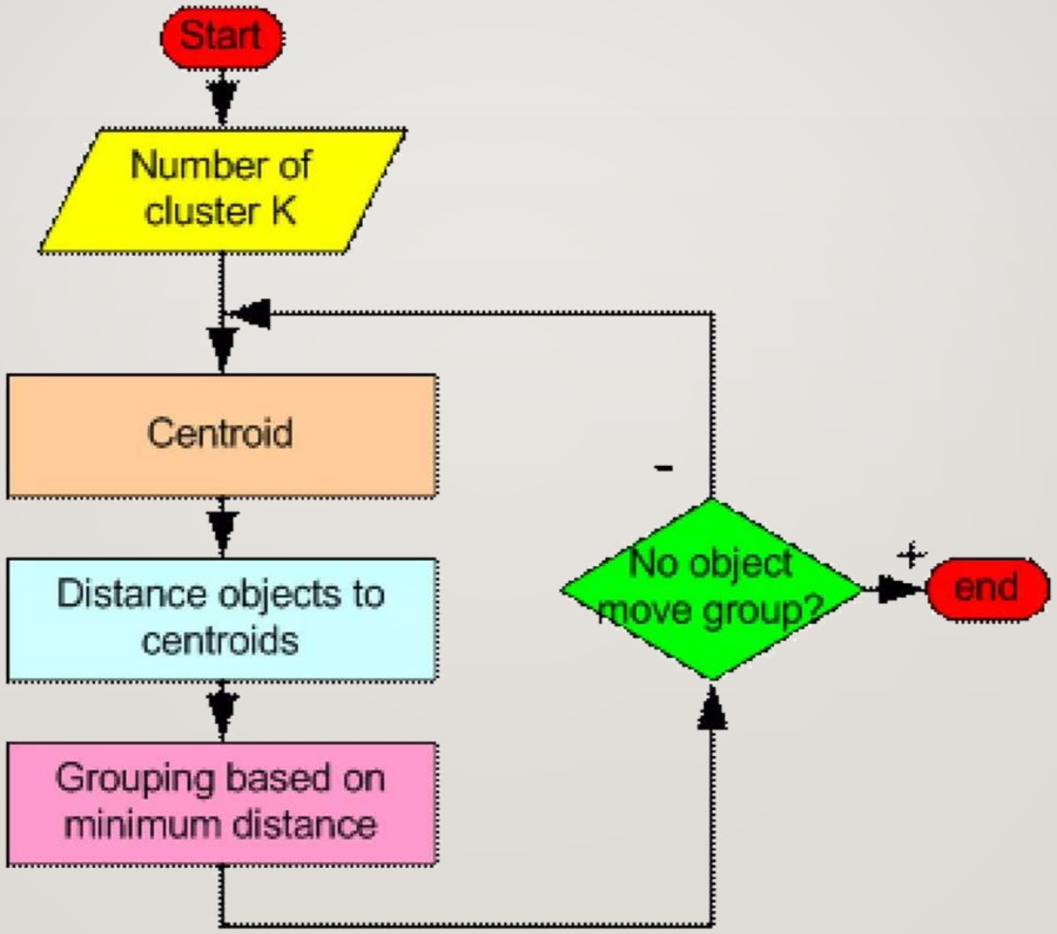
- An algorithm for partitioning (or clustering) N data points into K disjoint subsets S_j containing data points so as to minimize the sum-of-squares criterion

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} \left| x_n - \mu_j \right|^2,$$

where x_n is a vector representing the the n^{th} data point and u_j is the geometric centroid of the data points in S_j.

- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

How the K-Mean Clustering algorithm works?



A Simple example showing the implementation of k-means algorithm (using K=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 1:

Initialization: Randomly we choose following two centroids (k=2) for two clusters.

In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

- Thus, we obtain two clusters containing:

  {1,2,3} and {4,5,6,7}.

- Their new centroids are:

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

$$= (4.12, 5.38)$$

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

Step 3:

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.
- Therefore, the new clusters are:

  {1,2} and {**3**,4,5,6,7}
- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Step 4 :

  The clusters obtained are: {1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

(with K=3)

Individual	$m_1 = 1$	$m_2 = 2$	$m_3 = 3$	cluster
1	0	1.11	3.61	1
2	1.12	0	2.5	2
3	3.61	2.5	0	3
4	7.21	6.10	3.61	3
5	4.72	3.61	1.12	3
6	5.31	4.24	1.80	3
7	4.30	3.20	0.71	3

C_3

clustering with initial centroids (1, 2, 3)

Step 1

Individual	m_1 (1.0, 1.0)	m_2 (1.5, 2.0)	m_3 (3.9,5.1)	cluster
1	0	1.11	5.02	1
2	1.12	0	3.92	2
3	3.61	2.5	1.42	3
4	7.21	6.10	2.20	3
5	4.72	3.61	0.41	3
6	5.31	4.24	0.61	3
7	4.30	3.20	0.72	3

Step 2

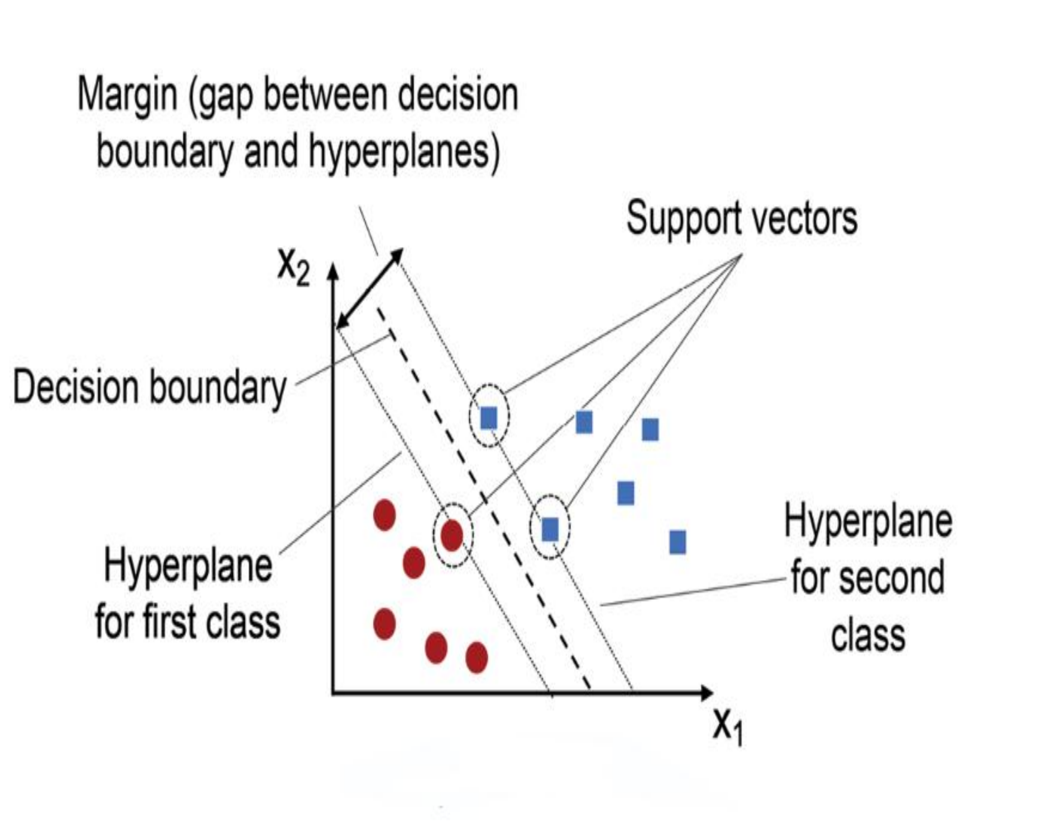
Weaknesses of K-Mean Clustering

1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The number of cluster, K, must be determined before hand. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments.
3. We never know the real cluster, using the same data, because if it is inputted in a different order it may produce different cluster if the number of data is few.
4. It is sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the *local optimum*.

WEEK 16
SUPPORT VECTOR MACHINE (SVM) ALGORITHM

SUPPORT VECTOR MACHINE (SVM) ALGORITHM

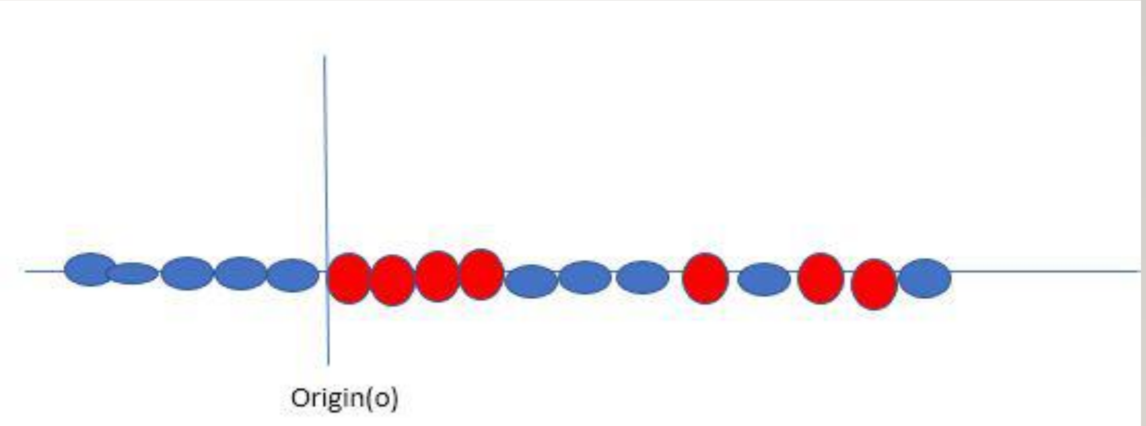
- A **Support Vector Machine (SVM)** is a powerful **machine learning algorithm** widely used for both **linear and nonlinear classification**, as well as **regression** and **outlier detection** tasks. SVMs are highly adaptable, making them suitable for various applications such as **text classification**, **image classification**, **spam detection**, **handwriting identification**, **gene expression analysis**, **face detection**, and **anomaly detection**.



HOW DOES SUPPORT VECTOR MACHINE ALGORITHM WORK?

- One reasonable choice for the **best hyperplane** in a **Support Vector Machine (SVM)** is the one that maximizes the **separation margin** between the two classes. The **maximum-margin hyperplane**, also referred to as the **hard margin**, is selected based on maximizing the distance between the hyperplane and the nearest data point on each side.

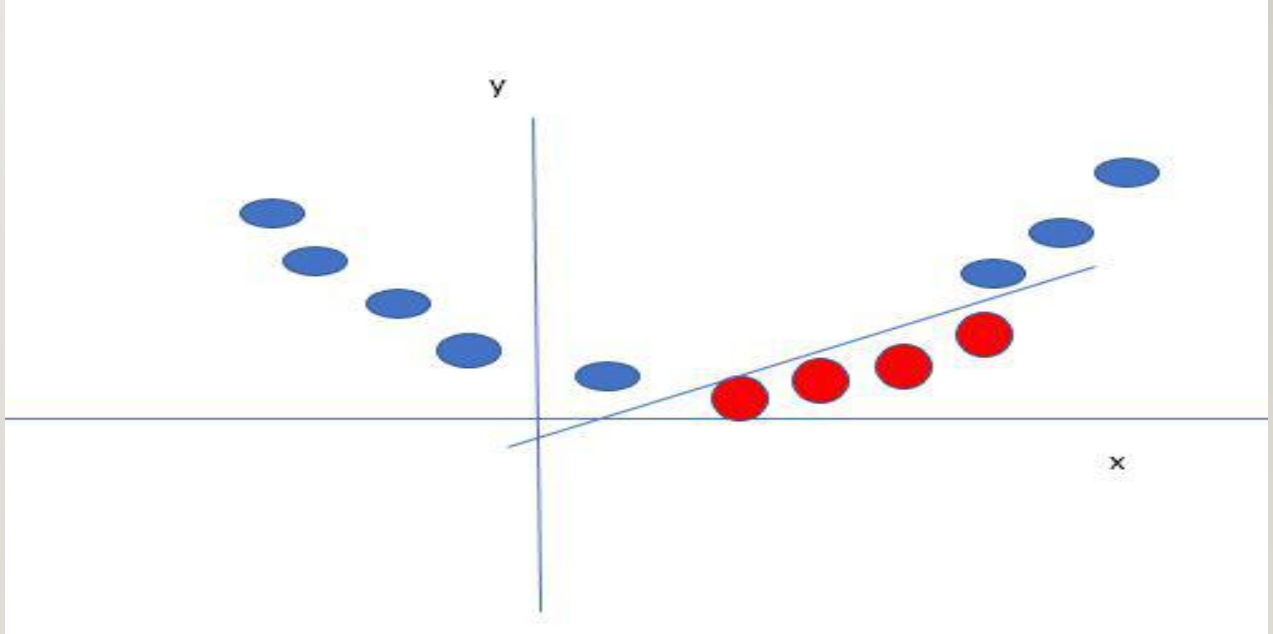
WHAT IF DATA IS NOT LINEARLY SEPARABLE???



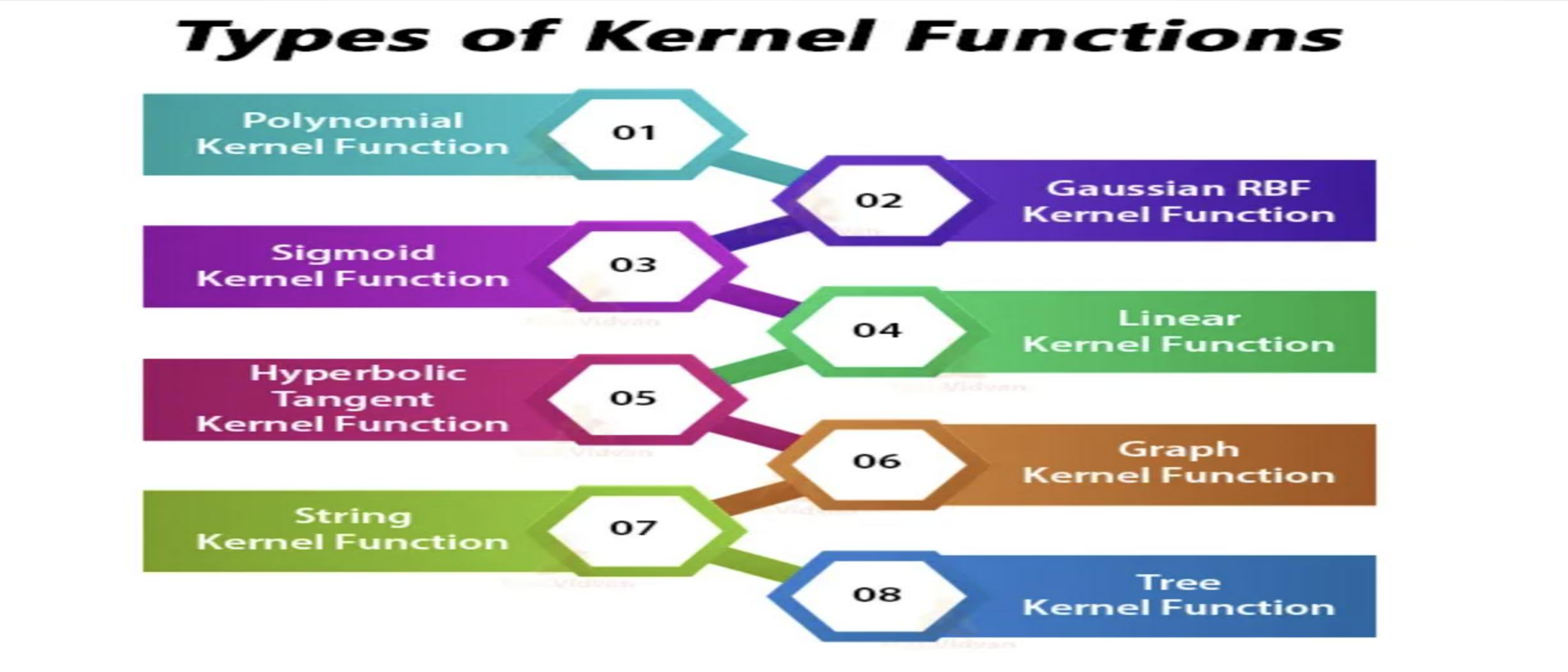
- Our data is shown in the figure above. SVM solves this by creating a new variable using a **kernel**. We call a point x_i on the line and we create a new variable y_i as a function of distance from origin o.so if we plot this we get something like as shown below

WHAT IF DATA IS NOT LINEARLY SEPARABLE???

- In this case, the new variable y is created as a function of distance from the origin. A non-linear function that creates a new variable is referred to as a kernel.



TYPES OF KERNEL IN SVM

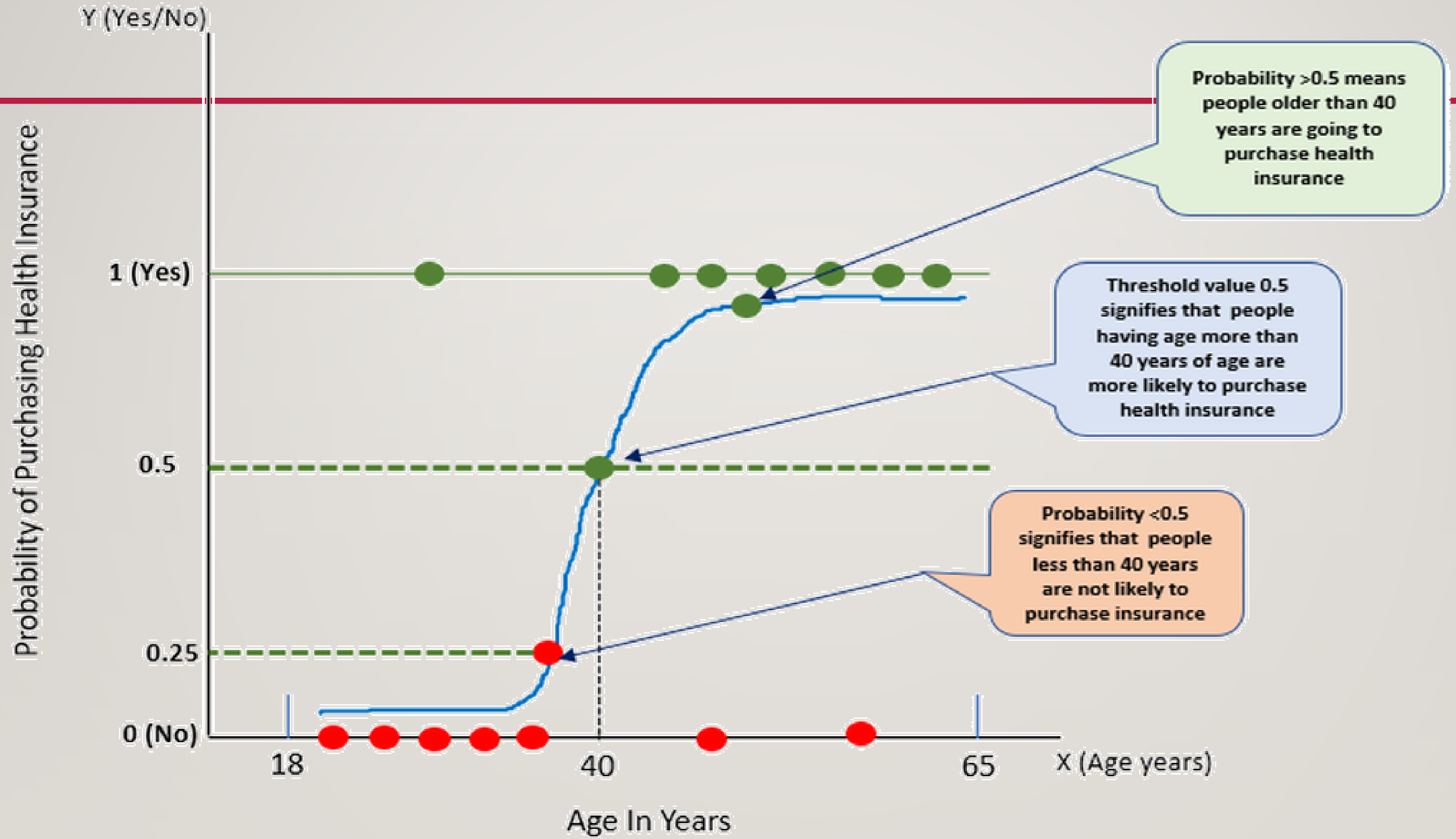


WEEK 17
LOGISTIC REGRESSION

LOGISTIC REGRESSION

- **Logistic regression** is a **supervised machine learning algorithm** used for **classification tasks** where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyze the relationship between two data factors.
- Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).

Logistic Regression Explained With Example !!!!!!



LINEAR VS LOGISTIC REGRESSION