MODULE 3 TOWARDS MACHINE LEARNING

What is machine learning?

- Machine learning is a type of artificial intelligence (AI) that allows systems to learn and improve from experience without being explicitly programmed
- Explicitly programming means telling the computers what to do by providing exact rules. If you are responsible to write a software, you can't leave a vague area, you need to give precise commands.

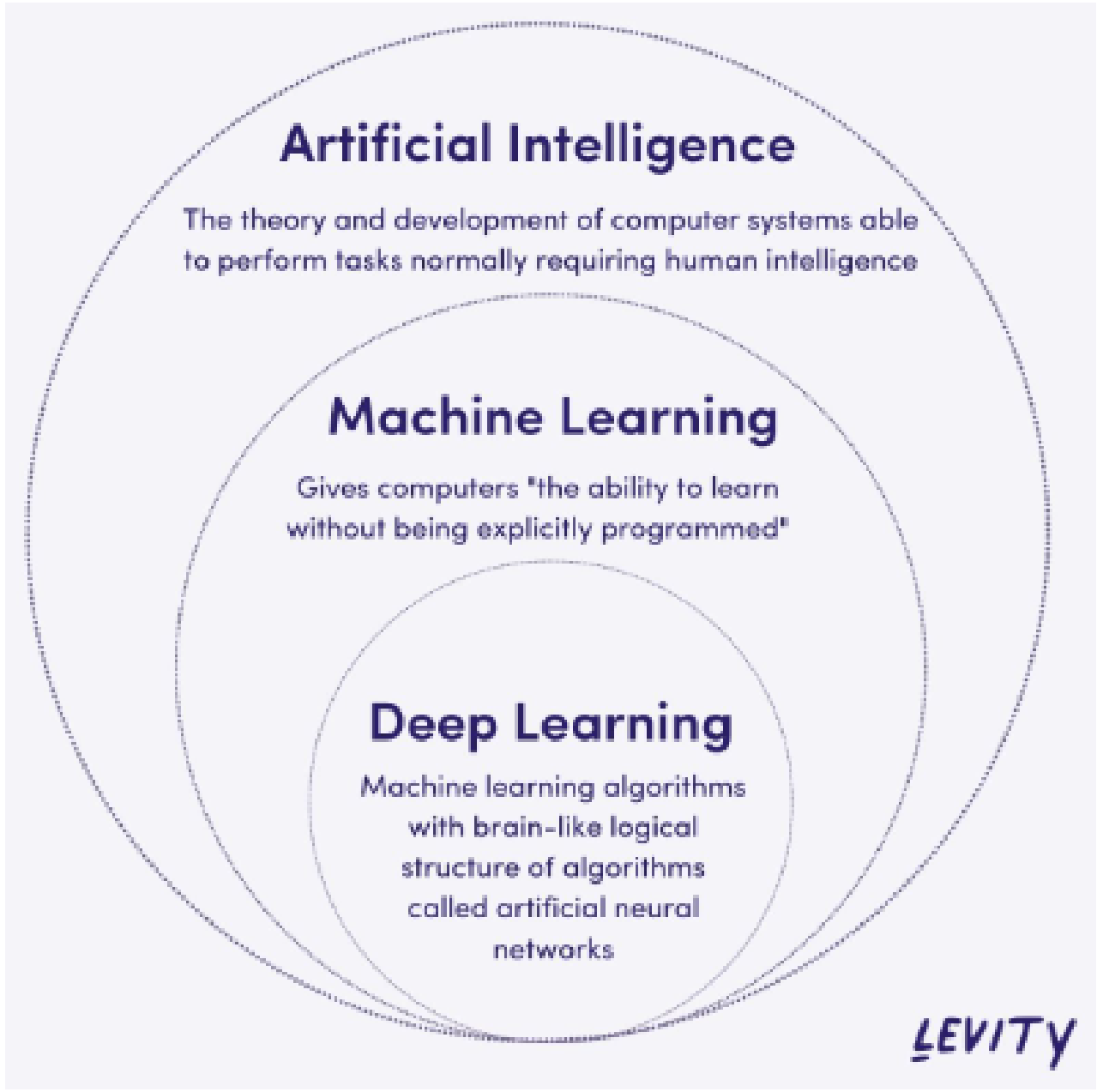
How it works?

- Machine learning uses algorithms to analyze large amounts of data, identify patterns, and make decisions. The more data a machine learning system is exposed to, the better it performs.

Key Concepts in Machine Learning:

- **Data**: Machine learning relies on large datasets to learn patterns and make predictions. Data can come in various forms, such as images, text, numbers, or any other structured or unstructured information.
- **Features**: Features are the measurable properties or characteristics of the data used by the model to learn patterns. For instance, in a dataset of houses, features might include square footage, number of rooms, or neighborhood.
- **Labels**: Labels are the target values or outputs that the model is trying to predict. For example, in a dataset predicting house prices, the label would be the price of each house.

- **Model:** A machine learning model is a mathematical representation that learns patterns from data. Once trained, it can make predictions or decisions without human intervention.
- **Training**: Training is the process of feeding data to the model and allowing it to adjust its parameters to minimize errors in its predictions.
- **Evaluation**: Once trained, models are tested on new data to assess their accuracy and generalizability.


Artificial Intelligence
The theory and development of computer systems able to perform tasks normally requiring human intelligence
Machine Learning
Gives computers "the ability to learn without being explicitly programmed"
Deep Learning
Machine learning algorithms with brain-like logical structure of algorithms called artificial neural networks
LEVITY

What is a Dataset ?

- A Dataset is a set of data grouped into a collection with which developers can work to meet their goals.
- In a dataset, the rows represent the number of data points and the columns represent the features of the Dataset.
- They are mostly used in fields like machine learning, business, and government to gain insights, make informed decisions, or train algorithms.

- This is the Iris dataset. Since this is a dataset with which we build models, there are input features and output features. Here:
- The input features are Sepal Length, Sepal Width, Petal Length, and Petal Width.
- Species is the output feature.

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

- Datasets can be stored in multiple formats. The most common ones are CSV, Excel, JSON, and zip files for large datasets such as image datasets.

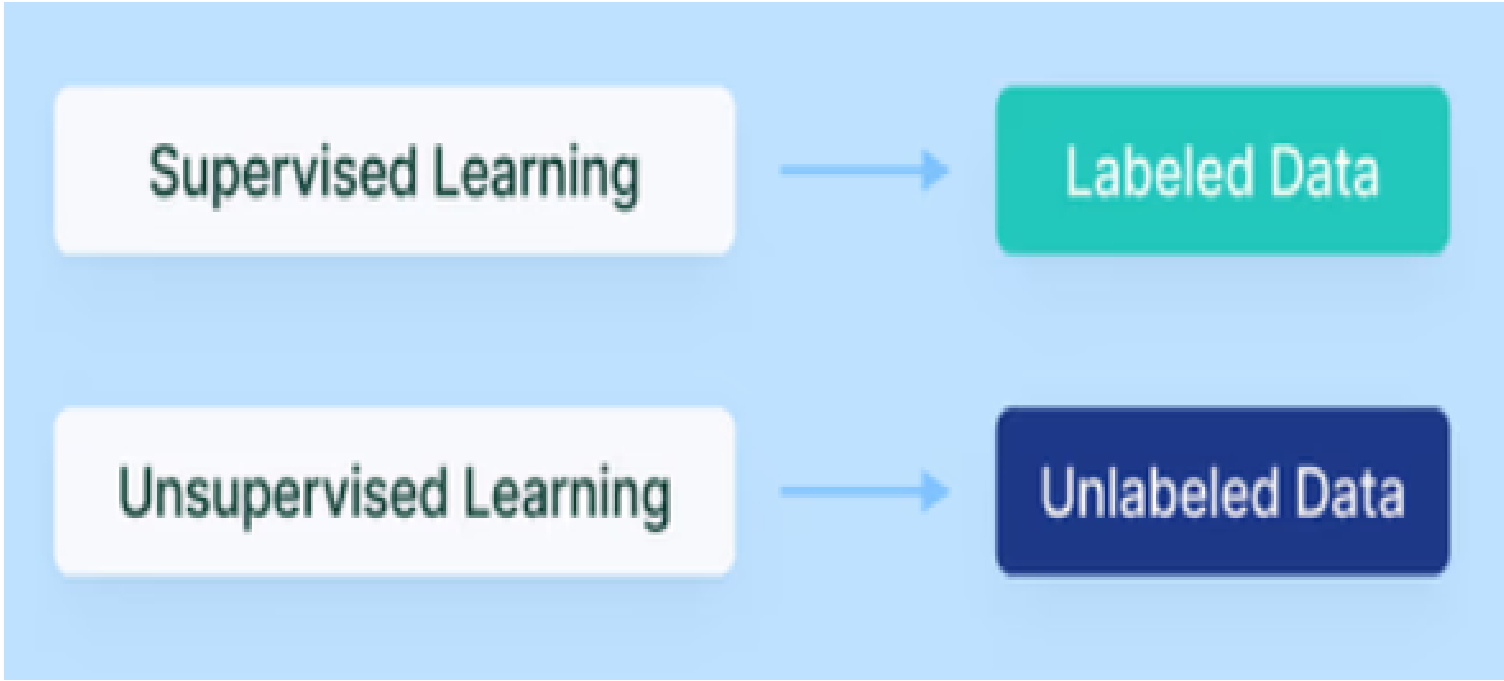
```
Name, Score, Department
Alex, 528, IT
Mallika, 650, Commerce
Joy, 670, Humanities
Yash, 679, IT
```

CSV

```
{ "name":"Thanos" }
```

JSON

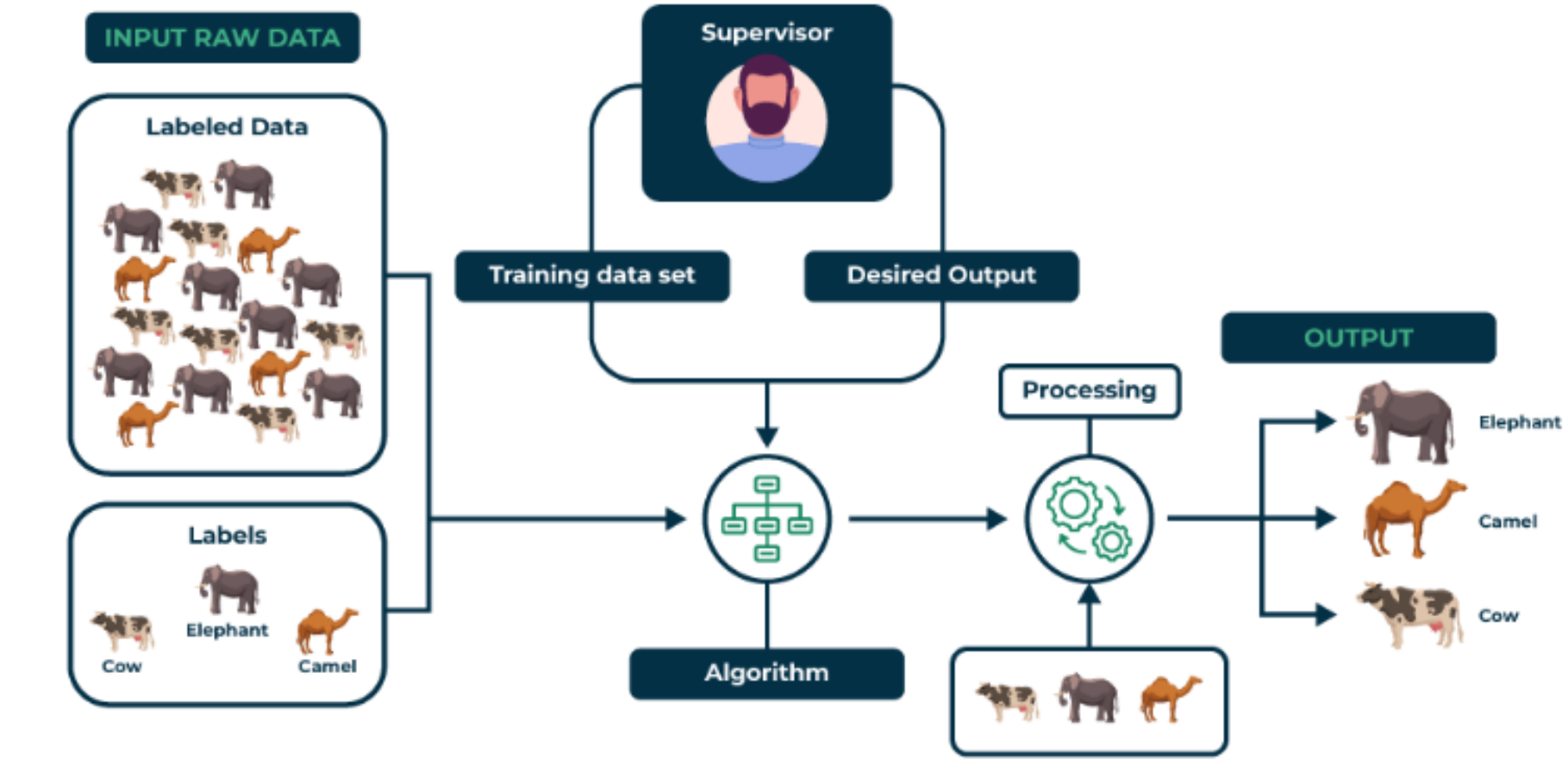
Types of Machine Learning:



What is 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, as the name indicates, has the presence of a supervisor as a teacher.
- Supervised learning is when we teach or train the machine using data that is well-labelled. Which means some data is already tagged with the correct answer.

Supervised Learning



Key Points:

- Supervised learning involves training a machine from labeled data.
- Labeled data consists of examples with the correct answer or classification.
- The machine learns the relationship between inputs and outputs .
- The trained machine can then make predictions on new, unlabeled data.

Types of Supervised Learning

- Supervised learning is classified into two categories of algorithms:
- **Regression:** A regression problem is when the output variable is a real value, such as “dollars” or “weight”.
- **Classification**: A classification problem is when the output variable is a category, such as “Red” or “blue” , “disease” or “no disease”.

Regression

- **Regression** is a type of supervised learning that is used to predict continuous values, such as house prices, stock prices, or customer churn.
- Regression algorithms learn a function that maps from the input features to the output value.
- **Some common regression algorithms include:**
- Linear Regression
- Polynomial Regression
- Support Vector Machine Regression

Classification

- **Classification** is a type of supervised learning that is used to predict categorical values,whether an email is spam or not, or whether a medical image shows a tumor or not
- **Some common classification algorithms include:**
- Logistic Regression
- Support Vector Machines
- Decision Trees
- Random Forests
- Naive Baye

Applications of Supervised learning

- **Supervised learning can be used to solve a wide variety of problems, including:**
- ***Spam filtering***: Supervised learning algorithms can be trained to identify and classify spam emails based on their content, helping users avoid unwanted messages.
- ***Image classification***: Supervised learning can automatically classify images into different categories, such as animals, objects, or scenes, facilitating tasks like image search, content moderation, and image-based product recommendations.

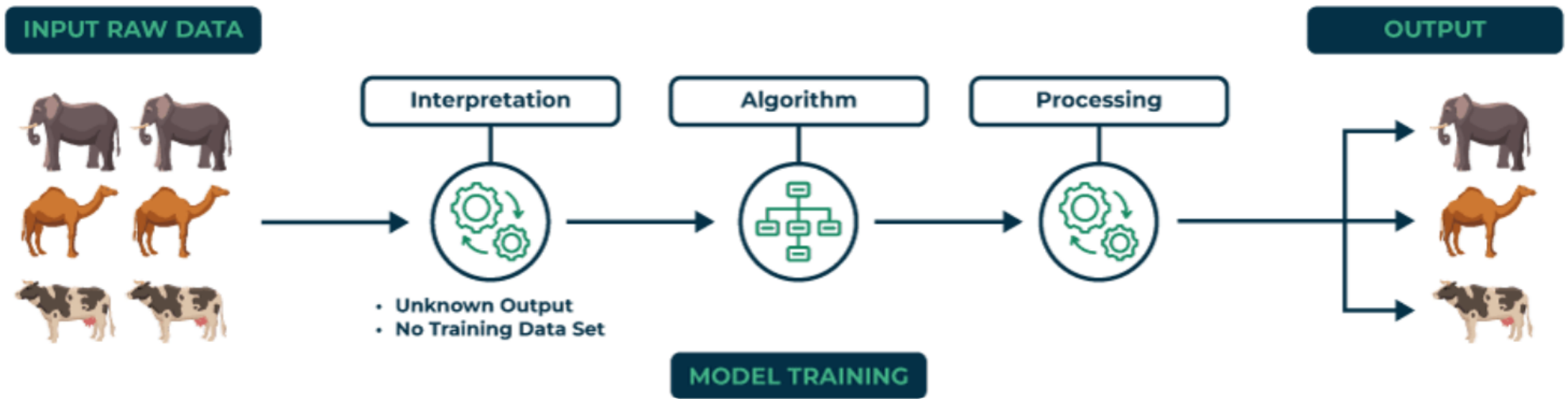
- ***Medical diagnosis***: Supervised learning can assist in medical diagnosis by analyzing patient data, such as medical images, test results, and patient history, to identify patterns that suggest specific diseases or conditions.
- ***Fraud detection***: Supervised learning models can analyze financial transactions and identify patterns that indicate fraudulent activity, helping financial institutions prevent fraud and protect their customers.

What is Unsupervised learning?

- **Unsupervised learning** is a type of machine learning that learns from unlabeled data.
- This means that the data does not have any pre-existing labels or categories.
- The goal of unsupervised learning is to discover patterns and relationships in the data without any explicit guidance.

- Unsupervised learning is the training of a machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance.
- Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.
- Unlike supervised learning, no teacher is provided

Unsupervised Learning



Types of Unsupervised Learning

- Unsupervised learning is classified into two categories of algorithms:
- ***Clustering:*** A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- ***Association:*** An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Clustering

- Clustering is a type of unsupervised learning that is used to group similar data points together.
- Clustering algorithms work by iteratively moving data points closer to their cluster centers and further away from data points in other clusters.
- **Clustering Types:-**
- Hierarchical clustering
- K-means clustering
- Principal Component Analysis

Application of Unsupervised learning

- ***Anomaly detection***: Unsupervised learning can identify unusual patterns or deviations from normal behavior in data, enabling the detection of fraud, intrusion, or system failures.
- ***Scientific discovery***: Unsupervised learning can uncover hidden relationships and patterns in scientific data, leading to new hypotheses and insights in various scientific fields.
- ***Recommendation systems***: Unsupervised learning can identify patterns and similarities in user behavior and preferences to recommend products, movies, or music that align with their interests.

Supervised learning	Unsupervised learning
Input data is labeled	Input data is unlabeled
Has a feedback mechanism	Has no feedback mechanism
Data is classified based on the training dataset	Assigns properties of given data to classify it
Divided into Regression & Classification	Divided into Clustering & Association
Used for prediction	Used for analysis
Algorithms include: decision trees, logistic regressions, support vector machine	Algorithms include: k-means clustering, hierarchical clustering, apriori algorithm
A known number of classes	A unknown number of classes

Splitting data – Explanation on Train/Test/ Validation sets.

- The train-validation-test split is fundamental to the development of robust and reliable machine learning models.
- Avoid employing the same dataset to train your machine learning model that you use to evaluate it.
- Doing so will result in a biased model that reports an artificially high model accuracy against the dataset it was trained on and poor model accuracy against any other dataset.

Training vs. Validation vs. Test Sets

Training Set

- The training set is the portion of the dataset reserved to fit the model. In other words, the model sees and learns from the data in the training set to directly improve its parameters.

- To maximize model performance, the training set must be
- (i) large enough to yield meaningful results (but not too large that the model overfits) and
- (ii) representative of the dataset as a whole. This will allow the trained model to predict any unseen data that may appear in the future.

Validation Set

- The validation set is the set of data used to evaluate and fine-tune a machine learning model during training, helping to assess the model's performance and make adjustments.
- By evaluating a trained model on the validation set, we gain insights into its ability to generalize to unseen data. This assessment helps identify potential issues such as overfitting, which can have a significant impact on the model's performance in real-world scenarios.

Test Set

- The test set is the set of data used to evaluate the final performance of a trained model.
- It serves as an unbiased measure of how well the model generalizes to unseen data, assessing its generalization capabilities in real-world scenarios.
- This assessment enables us to determine if the trained model has successfully learned relevant patterns and can make accurate predictions beyond the training and validation contexts.

1. Training Set

- **Purpose:** Used to train the model by adjusting its parameters.
- **Size:** Typically 60-70% of the dataset.

2. Validation Set

- **Purpose:** Used to tune hyperparameters and evaluate the model during training.
- **Size:** Typically 10-20% of the dataset

3. Test Set

- **Purpose:** Used to evaluate the final performance of the trained model.
- **Size:** Typically 20-30% of the dataset.

CONCEPT OF UNDERFITTING AND OVERFITTING

1. Bias

- **What it is:**
- Bias refers to the error introduced by approximating a real-world problem (which may be complex) with a simplified model.
- **Characteristics:**
  - A **high-bias model** makes strong assumptions about the data and oversimplifies the relationships.
  - A **low-bias model** is more flexible and capable of capturing complex patterns.

2. Variance

- **What it is:**
- Variance measures how much the model's predictions change when trained on different subsets of the training data. It captures the model's sensitivity to fluctuations in the training set.
- **Characteristics:**
  - A **high-variance model** is overly complex and closely follows the training data.
  - A **low-variance model** is more consistent across different datasets but might not capture all patterns.

Bias Variance Tradeoff

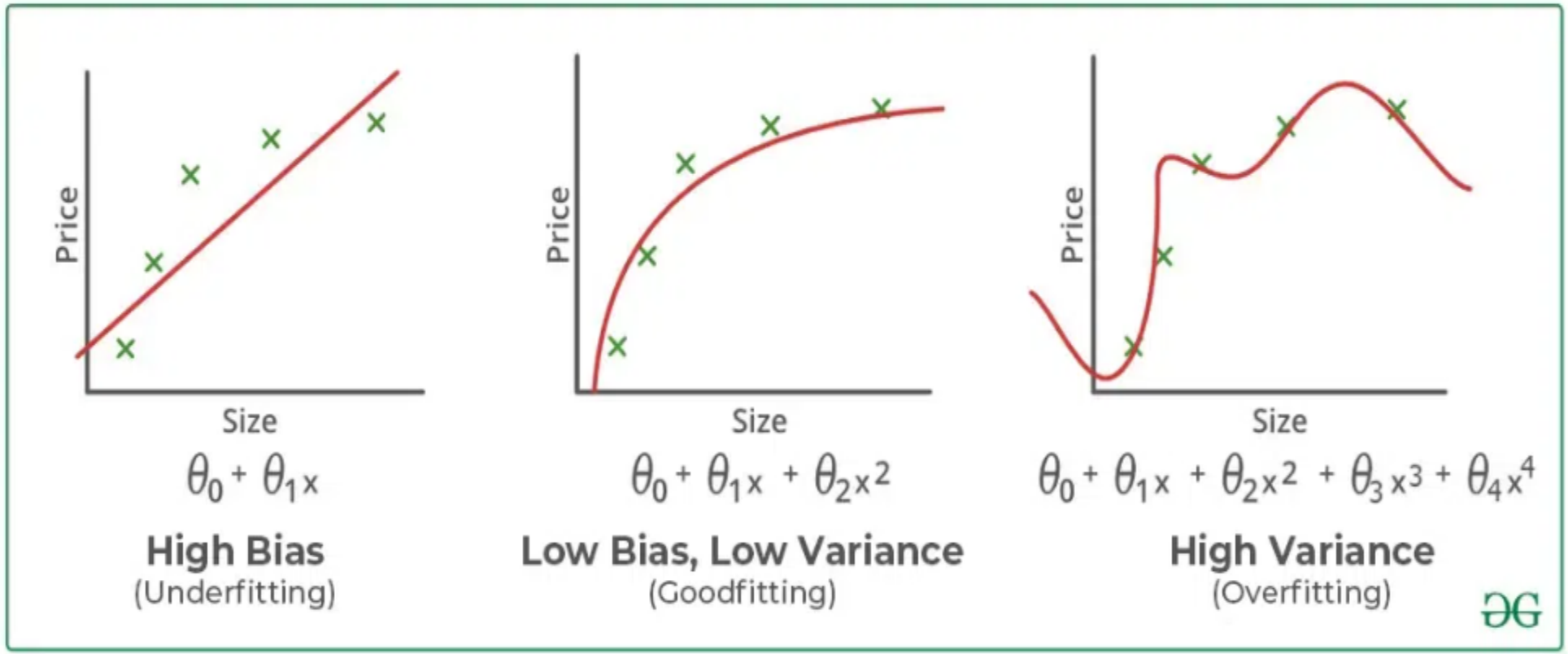
- If the algorithm is too simple then it may be on high bias and low variance condition and thus is error-prone.
- If algorithms fit too complex then it may be on high variance and low bias.
- In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off.
- An algorithm can’t be more complex and less complex at the same time.

- In machine learning, the goal is to find a balance between bias and variance to minimize **total error** on unseen data. The total error can be expressed as:

$$\textbf{Total Error} = \textbf{Bias}^2 + \textbf{Variance} + \textbf{Irreducible Error}$$

or

- **Bias** contributes to error because the model is too simple (underfitting).
- **Variance** contributes to error because the model is too complex (overfitting).
- **Irreducible Error** is noise inherent to the data that cannot be eliminated.


Price
Size
$\theta_0 + \theta_1 x$
High Bias
(Underfitting)
Price
Size
$\theta_0 + \theta_1 x + \theta_2 x^2$
Low Bias, Low Variance
(Goodfitting)
Price
Size
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$
High Variance
(Overfitting)

Underfitting in Machine Learning

- A statistical model or a machine learning algorithm is said to have underfitting when a model is too simple to capture data complexities.
- It represents the inability of the model to learn the training data effectively result in poor performance both on the training and testing data.
- In simple terms, an underfit model’s are inaccurate, especially when applied to new, unseen examples. It mainly happens when we uses very simple model with overly simplified assumptions. .

- To address underfitting problem of the model, we need to use more complex models, with enhanced feature representation, and less regularization
- ***Note: The underfitting model has High bias and low variance***.

Reasons for Underfitting

- The model is too simple, So it may be not capable to represent the complexities in the data.
- The input features which is used to train the model is not the adequate representations of underlying factors influencing the target variable.
- The size of the training dataset used is not enough.
- Excessive regularization are used to prevent the overfitting, which constraint the model to capture the data well.
- Features are not scaled.

Techniques to Reduce Underfitting

- Increase model complexity.
- Increase the number of features.
- Remove noise from the data.
- Increase the number of epochs or increase the duration of training to get better results.

Overfitting in Machine Learning

- A statistical model is said to be overfitted when the model does not make accurate predictions on testing data.
- When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set.
- And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise.

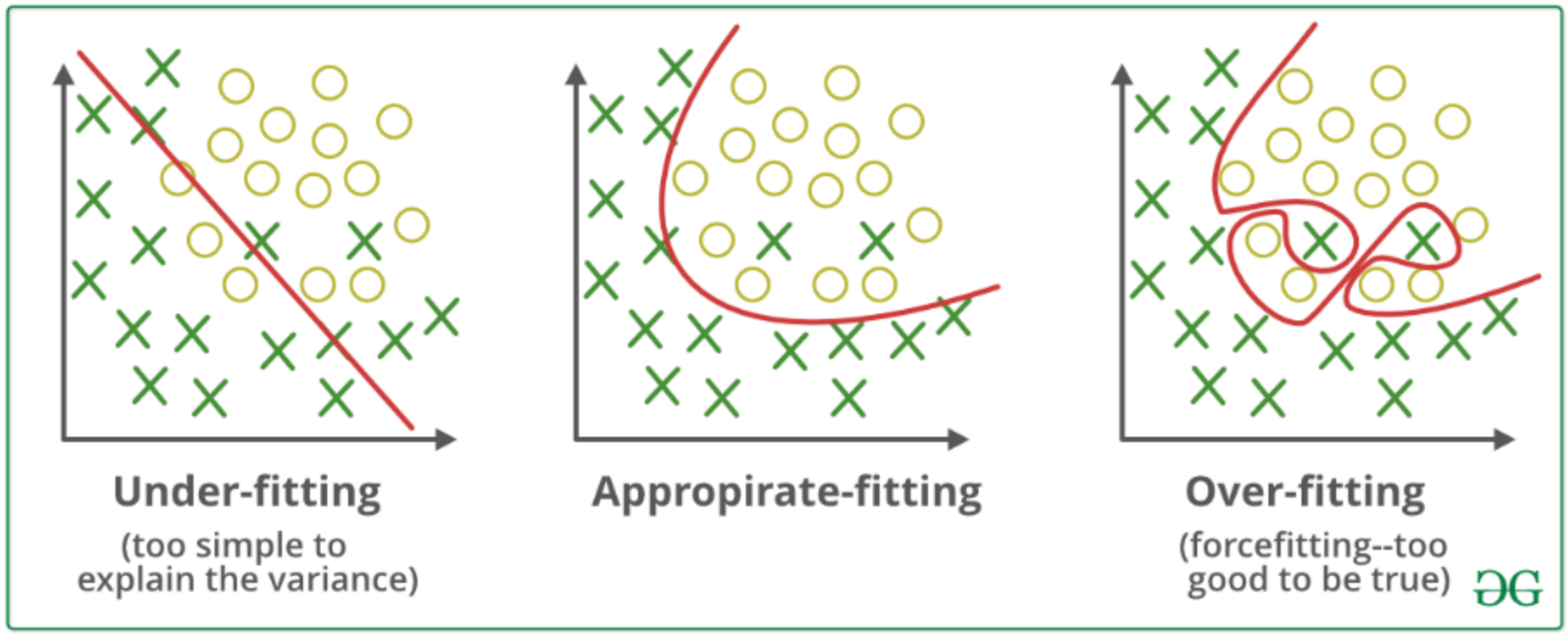
- The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models.
- A solution to avoid overfitting is using a linear algorithm if we have linear data

Reasons for Overfitting:

- High variance and low bias.
- The model is too complex.
- The size of the training data.

Techniques to Reduce Overfitting

- Improving the quality of training data reduces overfitting by focusing on meaningful patterns, mitigate the risk of fitting the noise or irrelevant features.
- Increase the training data can improve the model's ability to generalize to unseen data and reduce the likelihood of overfitting.
- Reduce model complexity.
- Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
- Ridge Regularization and Lasso Regularization.
- Use dropout for neural networks to tackle overfitting.


Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)

Modeling Input and Output: Vectors and Presenting Pictures

- In machine learning and data science, modeling input and output using vectors and images is crucial.
- **1. Vectors as Inputs and Outputs**
- A **vector** is simply a one-dimensional array of numbers, often used to represent data in a machine-readable format.
- In modeling:
  - **Inputs:** Represent features (or attributes) of the data.
  - **Outputs:** Represent predictions, classifications, or decisions made by the model.

- **Examples:**
- **Numerical Data:**
  - A vector could represent a customer's age, income, and spending score in a shopping analysis: $x=[35,50000,72]$
  - This would be the input for the model.
- **Output Example:**
  - If the task is classification (e.g., whether a customer will buy a product), the output could be:

$$y=1(\text{Yes}) \quad \text{or} \quad y=0(\text{No}).$$

- 2. Modeling Pictures as Vectors
- An image is essentially a grid of tiny squares called **pixels**. Each pixel has a numerical value representing its color or intensity.

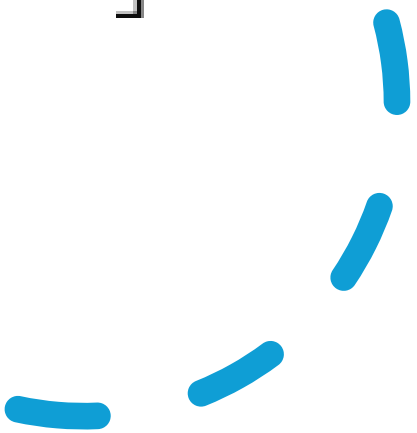
  - **Grayscale Images:** Each pixel is represented by a single value, typically ranging from 0 (black) to 255 (white).
  - **Color Images:** Each pixel is represented by three values (one for each channel: Red, Green, Blue), each ranging from 0 to 255.

Example:

- A 4×4 grayscale image:

$$\begin{bmatrix} 0 & 255 & 128 & 64 \\ 32 & 64 & 128 & 255 \\ 255 & 0 & 64 & 128 \\ 128 & 64 & 0 & 255 \end{bmatrix}$$

Each number represents the brightness of a pixel.

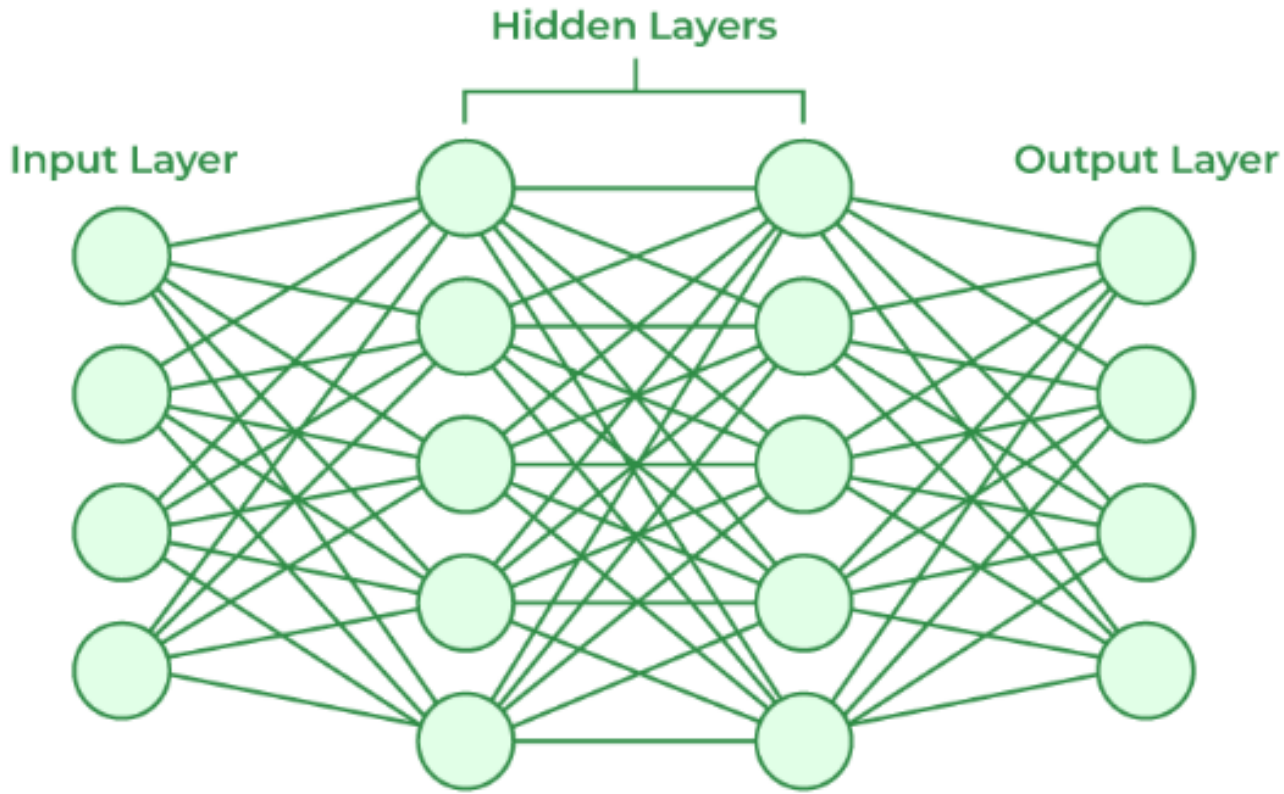


- **Flattening the Image into a Vector**
- To use the image as input for a machine learning model, it is **flattened** into a one-dimensional vector:

Example:

- A 4×4 grayscale image (16 pixels total) is flattened into:

$$\mathbf{x} = [0, 255, 128, 64, 32, 64, 128, 255, 255, 0, 64, 128, 128, 64, 0, 255]$$



Neural Networks Architecture

What Happens Inside the Model

Input Layer:

- The vector representation of the data is fed into the model.

Hidden Layers:

- The model performs mathematical operations like:
  - **Dot Products** between input vectors and weights.
  - **Non-linear transformations** using activation functions.

Output Layer:

- The output is typically another vector:
  - Regression: Single numeric value (e.g., house price).
  - Classification: Probability vector (e.g., [0.8,0.2][0.8, 0.2][0.8,0.2] for "cat" vs. "dog").

Stochastic and Deterministic Training

- Training a machine learning model involves adjusting its parameters to minimize the error or loss. This process can be approached in two main ways:
- **stochastic training** and
- **deterministic training**.

Stochastic Training

- Stochastic training refers to a training approach in machine learning where **randomness** is intentionally introduced during the training process
- This randomness plays a crucial role in improving efficiency and sometimes helps the model generalize better
- Stochastic training is commonly used in large-scale machine learning tasks

How Stochastic Training Works

- Randomness can be introduced in several ways during training:
- **Random Sampling of Data (Mini-Batches):**
  - Instead of processing the entire dataset at once (as in full-batch training), the model uses **randomly selected subsets of data** called **mini-batches** during each training step.
  - This reduces computational cost and makes training feasible for large datasets.
- **Random Initialization of Parameters:**
  - The weights of the model (e.g., in a neural network) are often initialized randomly.
  - This avoids issues like symmetry and ensures that each run starts from a different point in the optimization landscape.

- **Data Augmentation with Randomness:**
  - Random transformations (e.g., rotation, flipping, cropping) are applied to images or other input data to artificially expand the training dataset and introduce variability.
- **Dropout (in Neural Networks):**
  - During training, some neurons are randomly "dropped" or deactivated, forcing the model to learn more robust features.

Deterministic Training

- Deterministic training ensures that the training process follows a **fixed, predictable path** without introducing any randomness
- Every aspect of the training process is controlled to make it **reproducible**: running the same model with the same dataset and hyperparameters will always yield the same results

How Deterministic Training Works

- **Uses the entire dataset for every update.**

Fixed Initialization of Parameters:

- Model parameters (e.g., weights and biases in neural networks) are initialized with the same fixed values every time training starts. This avoids variability caused by random initialization.

No Stochastic Elements:

- Regularization methods like **dropout** or **data augmentation** (which typically involve randomness) are avoided