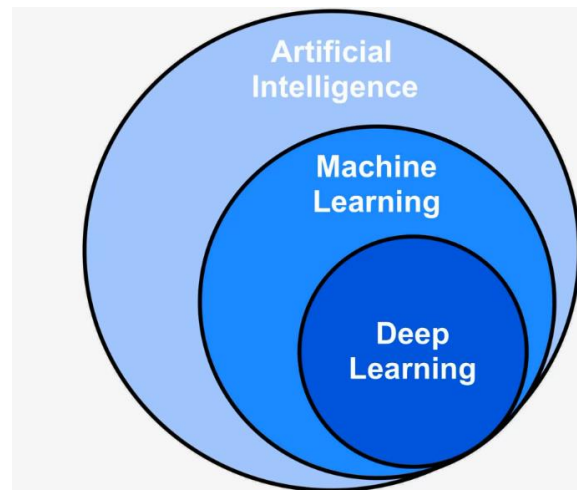
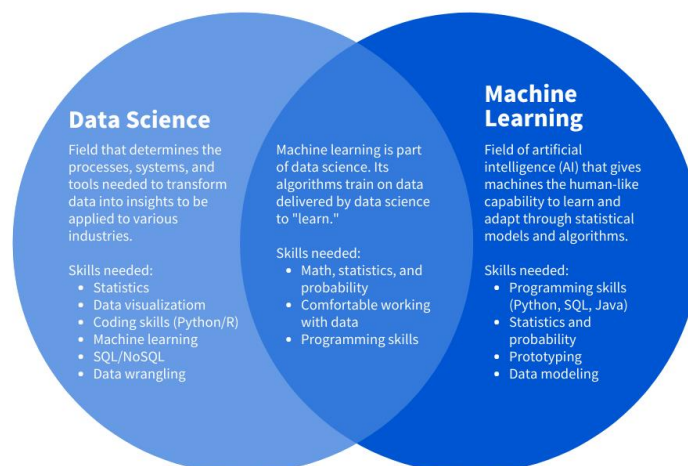


Machine Learning Fundamentals Roadmap



Machine Learning (ML) is a branch of artificial intelligence (AI) that focuses on building systems that can learn from data, identify patterns, and make decisions with minimal human intervention. Essentially, instead of hardcoding instructions, ML allows a model to learn from examples.

- **Artificial Intelligence (AI):** AI is the broader concept where machines are made to mimic human intelligence and behaviour. It includes many things like reasoning, problem-solving, learning, and perception.
- **Machine Learning (ML):** ML is a subset of AI that focuses specifically on how computers can learn from data without being explicitly programmed for every task. It's like teaching a machine to learn from experience.
- **Deep Learning (DL):** Deep Learning is a further subset of ML. It involves using complex models called neural networks that are inspired by the human brain. Deep Learning models can handle large datasets and learn intricate patterns, making them useful for tasks like image recognition and natural language processing (NLP).



Applications:

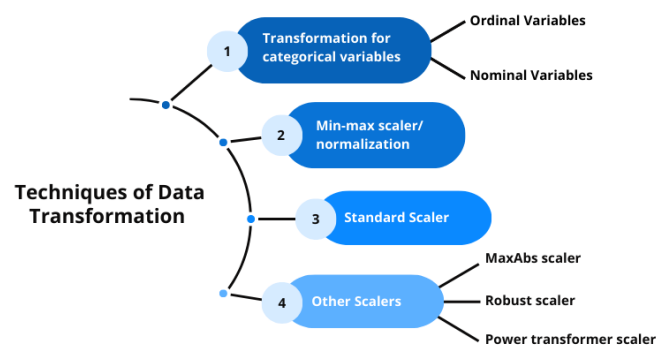
Machine Learning is used in many places in the real world. Some examples include:

- **Recommendation Systems:** Platforms like Netflix and YouTube use ML to recommend movies, shows, or videos based on what you've watched previously.
- **Image Recognition:** ML models help apps like Google Photos or Facebook automatically tag people in pictures, or in medical fields to detect diseases from medical images like X-rays.
- **Natural Language Processing (NLP):** This allows machines to understand, interpret, and generate human language. It's used in chatbots, virtual assistants like Siri or Alexa, and for translating languages (e.g., Google Translate).

2. Mathematics for Machine Learning (Prerequisites)

- **Linear Algebra:**
 - Vectors and Matrices
 - Matrix Operations (Addition, Multiplication)
 - Eigenvalues and Eigenvectors
 - Singular Value Decomposition (SVD)
- **Probability and Statistics:**
 - Probability Distributions (Gaussian, Bernoulli, Binomial, Poisson, etc.)
 - Descriptive Statistics (Mean, Median, Mode, Variance, etc.)
 - Bayes Theorem and Conditional Probability

3. Data Preprocessing



Before using machine learning algorithms, it's important to clean the data because real-world data is often incomplete, noisy, or inconsistent.

1. Handling Missing Data:

Missing data can cause issues in models, so it must be handled properly.

- **Imputation:** Replacing missing values with some meaningful number. Common techniques include using the mean, median, or mode of the data to fill in the gaps.
- **Dropping:** If too many values are missing or if the data point is not crucial, you can just remove the entire row or column with missing values.

2. Outlier Detection and Treatment:

Outliers are extreme values that are very different from the rest of the data. They can skew your model's performance if not handled.

- **Detection:** Methods like box plots or standard deviation help identify outliers.
- **Treatment:** You can either remove the outliers or transform the data (e.g., using logarithmic transformations) to reduce the effect of outliers.

3. Data Transformation

Transforming data is crucial to ensure that it is in a suitable format for machine learning algorithms.

- **Normalization:** This scales data to fit within a specific range, usually between 0 and 1. It is helpful when your features have different units or scales.
- **Standardization:** This scales data to have a mean of 0 and a standard deviation of 1. It's useful when the data follows a Gaussian (normal) distribution.
- **Logarithmic Transformation:** Used when you have highly skewed data. It compresses large values while preserving relationships between smaller values.

4. Feature Scaling:

- **Min-Max Scaling:** It scales all features to a range between 0 and 1. This is useful when you want all your features to have the same weight.
- **Z-score Normalization:** It transforms data into a distribution with a mean of 0 and standard deviation of 1. It's helpful when data is normally distributed and you want to standardize it.

5. Feature Engineering

Feature Engineering is about creating new features or transforming existing ones to make your model more effective. Features are the input data that the model uses to make predictions.

- **Encoding Categorical Variables:**

- **One-hot Encoding:** This turns categorical values (like "red," "blue," "green") into binary (0/1) values. For example, if you have a "colour" feature, one-hot encoding creates a separate column for each colour. [1, 0, 0] for Red, [0, 1, 0] for Blue, [0, 0, 1] for Green.

- **Label Encoding:** This replaces categories with numbers. Instead of creating new columns like one-hot encoding, it simply assigns a unique number to each category.

Example: Red = 1, Blue = 2, Green = 3.

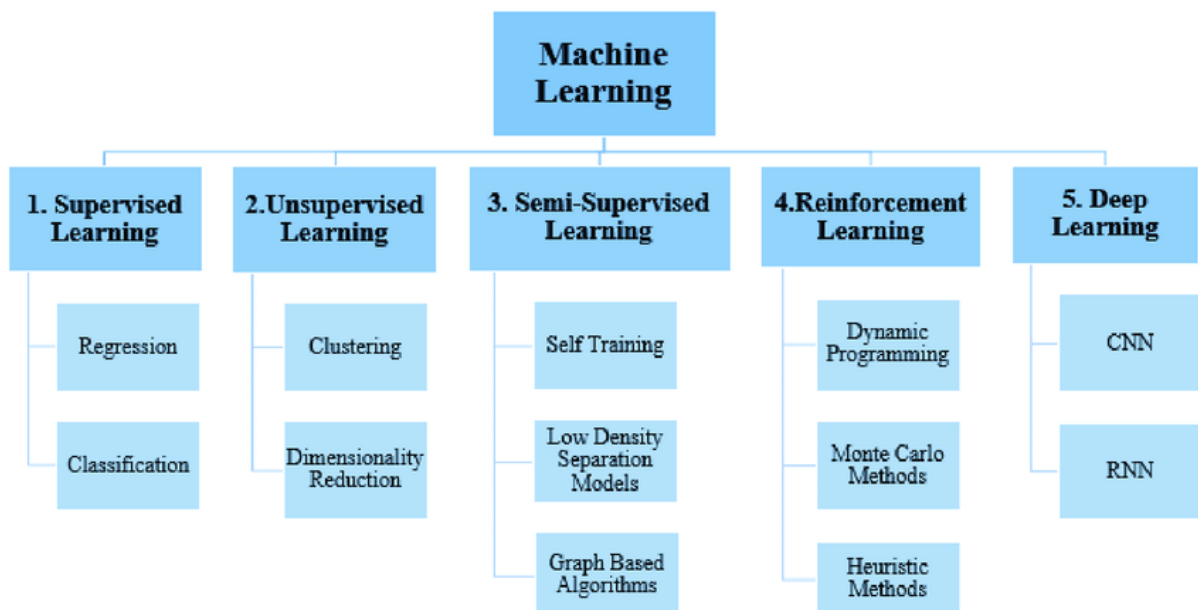
6. Dimensionality Reduction:

- **PCA (Principal Component Analysis):** This reduces the number of features while keeping as much of the original information as possible. It transforms the data into a smaller set of variables (called principal components).
- **LDA (Linear Discriminant Analysis):** This is similar to PCA but is used when the target variable is categorical. LDA tries to find the feature combinations that best separate the categories.

4. Types of Learning

Types of Machine Learning

There are several types of machine learning, and the type used depends on how the data is structured and the nature of the problem you're trying to solve. Let's dive into the main types:



1. Supervised Learning

- **Definition:**

In supervised learning, the machine is trained on a labelled dataset, meaning each input data point comes with a corresponding output or label. The goal is for the model to learn from the labelled data and make predictions for new, unseen data.

- **How it works:**

Imagine you have a dataset of houses with features like size, number of bedrooms, and price. The machine learns from this data (input: house features, output: price) and can then predict the price of a new house based on its features.

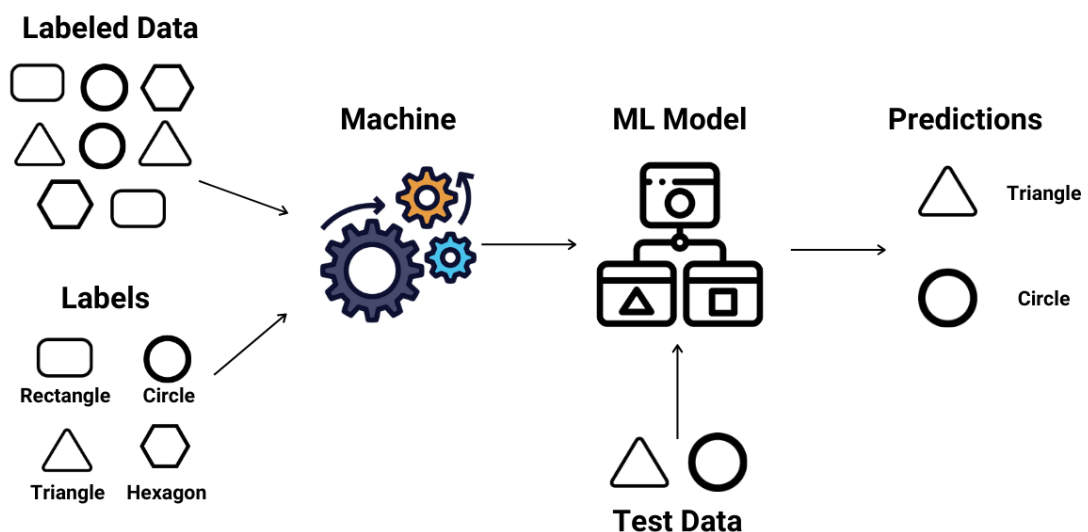
- **Example:**

- **Classification:** Predicting if an email is spam or not spam (binary labels).
- **Regression:** Predicting the price of a house (continuous labels).

- **Common Algorithms:**

- Linear Regression
- Decision Trees
- Support Vector Machines (SVM)
- Neural Networks

Supervised Learning



2. Unsupervised Learning

- **Definition:**

In unsupervised learning, the machine is given data without labels. The task is to discover hidden patterns or structures in the data. There's no "correct answer" to guide the model; it has to figure things out on its own.

- **How it works:**

Think of it like giving the machine a pile of unsorted photos and asking it to group similar ones together. It doesn't know what each photo represents (no labels), but it can identify that some images are similar and should belong to the same group.

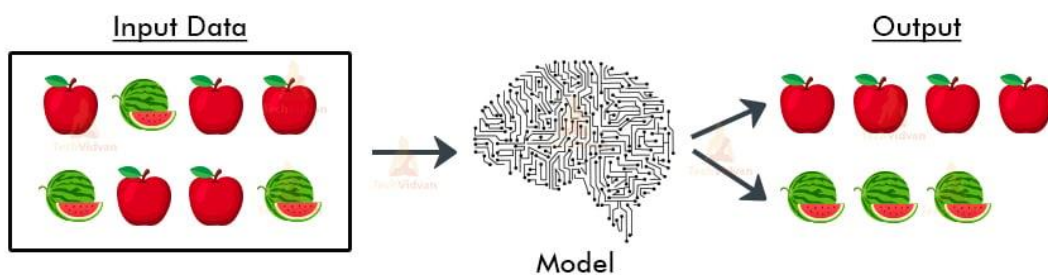
- **Example:**

- **Clustering:** Grouping customers with similar purchasing behaviour for targeted marketing.
- **Anomaly Detection:** Identifying fraudulent transactions in a dataset of financial records by spotting unusual patterns.

- **Common Algorithms:**

- K-Means Clustering
- Hierarchical Clustering
- Principal Component Analysis (PCA)

Unsupervised Learning in ML



3. Semi-supervised Learning

- **Definition:**

Semi-supervised learning is a mix of supervised and unsupervised learning. In this approach, the model is trained on a small amount of labelled data and a large amount of unlabelled data. This is useful when labelling data is expensive or time-consuming, but you still want to make use of all available data.

- **How it works:**

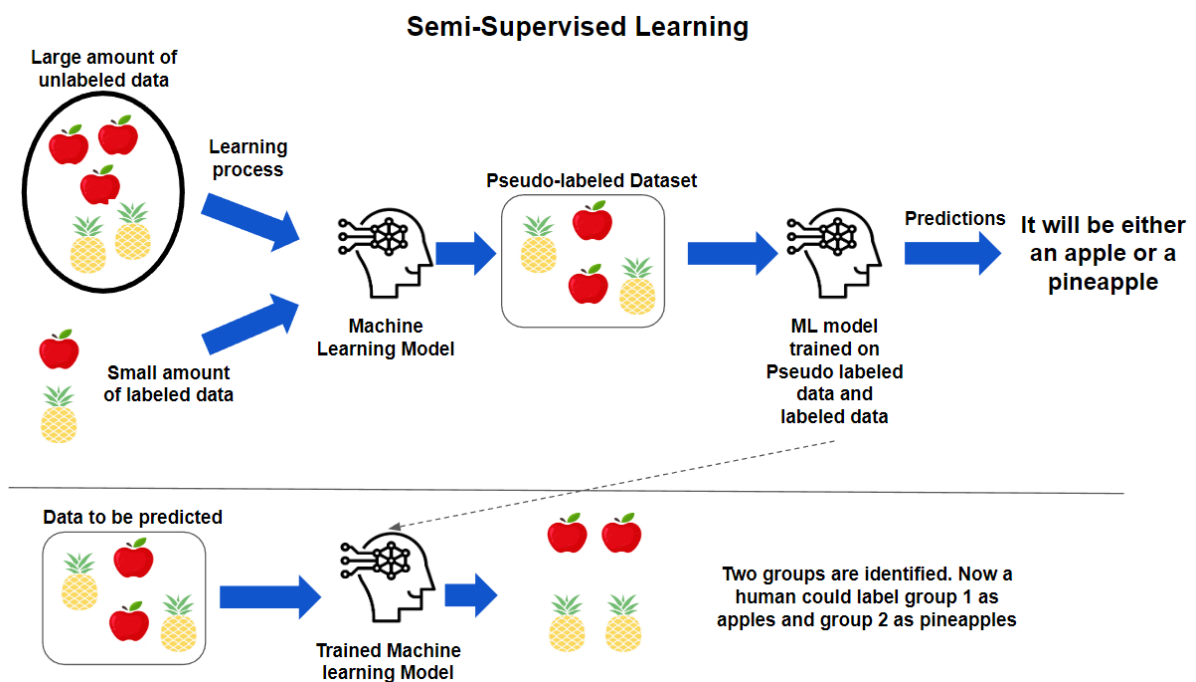
Let's say you're building a model to recognize different animals in images. Labelling thousands of images might be hard, but if you label only a few (like 100) and then let the machine learn from the rest of the unlabelled images, it can improve its performance.

- **Example:**

- A dataset with 1,000 photos, but only 100 are labelled. The model learns from both the labelled and unlabelled photos to improve its ability to identify new photos.

- **Common Algorithms:**

- Self-training
- Co-training
- Semi-supervised SVMs



4. Reinforcement Learning

- **Definition:**

In reinforcement learning, the machine learns by interacting with an environment. It takes actions and receives feedback in the form of rewards or penalties. The goal is to learn a strategy or policy that maximizes the total reward over time.

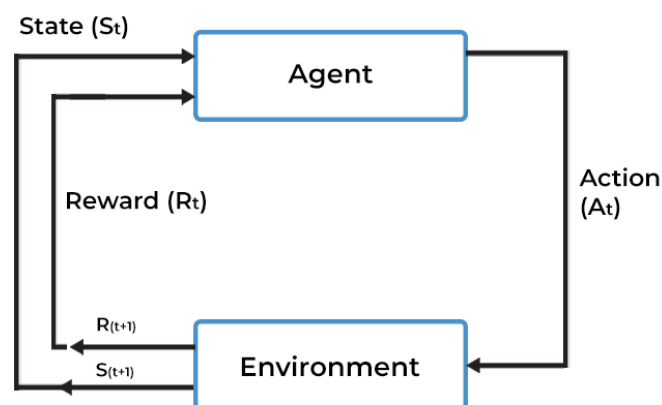
- **How it works:**

Imagine training a robot to navigate a maze. The robot moves in the maze and gets positive rewards for moving towards the exit and negative rewards (penalties) for hitting walls. Over time, it learns the best actions to take at each step to get out of the maze faster.

- **Example:**

- Training an AI to play a game like chess, where it learns from each move whether it's getting closer to winning or losing.
- Self-driving cars learning to make driving decisions (like turning or stopping) based on traffic rules and conditions.

REINFORCEMENT LEARNING MODEL



In machine learning, **regression** and **classification** are two fundamental types of tasks. Both deal with making predictions, but they do so in different ways:

- **Regression:** Predicts continuous values (e.g., predicting house prices, temperature, etc.).
- **Classification:** Predicts discrete labels or categories (e.g., predicting whether an email is spam or not).

5. Regression

1. Linear Regression

- **Definition:**

Linear regression is the simplest form of regression, where the model assumes a straight-line relationship between the input (features) and the output (target). The goal is to find the line that best fits the data.

- **How it works:**

It predicts the target (Y) using a formula:

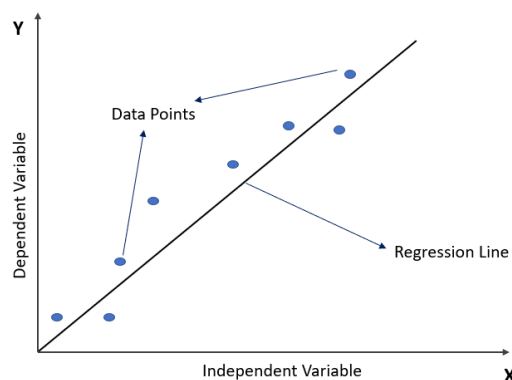
$$Y = mX + b$$

- **Where:**

- X is the input data,
- m is the slope of the line (how steep the line is),
- b is the intercept (where the line crosses the Y-axis).

- **Example:**

Predicting house prices based on the size of the house. A bigger house would usually have a higher price, and a linear regression model can draw a line through the data to predict prices.



2. Polynomial Regression

- **Definition:**

Polynomial regression fits a curve (rather than a straight line) to the data by including powers of the input features. It's used when the relationship between the input and output is non-linear.

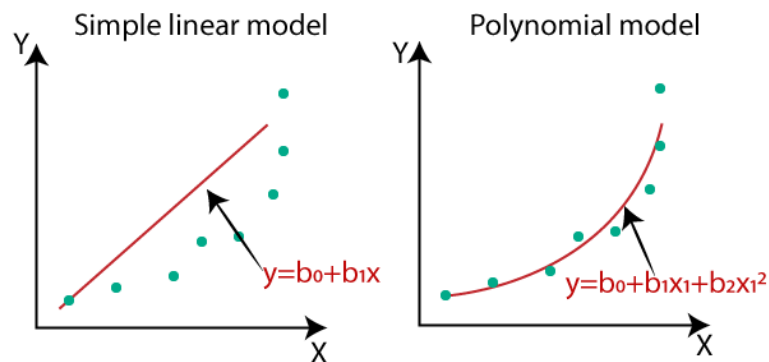
- **How it works:**

The model takes the input XXX and raises it to different powers (like X^2 , X^3 , X^2 , X^3) to fit more complex data patterns.

- **Example:**

Predicting the growth of a population over time, where growth doesn't happen at a constant

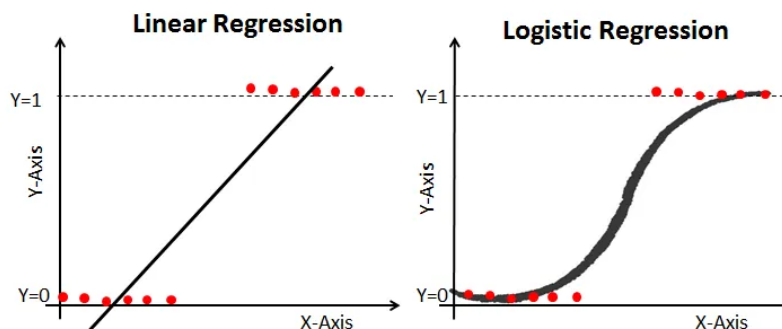
rate but speeds up over time. A polynomial curve fits this type of data better than a straight line.



6. Classification

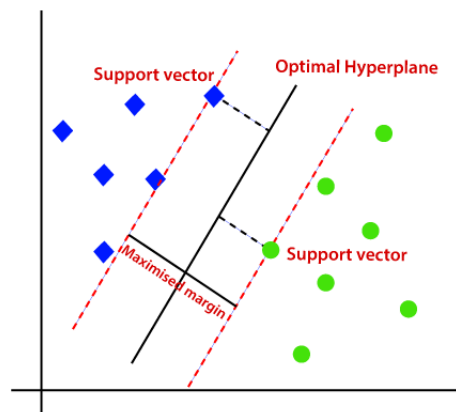
1. Logistic Regression

- **Definition:**
Despite its name, logistic regression is used for classification, not regression. It predicts the probability of a binary outcome (like yes/no, 0/1). It's often used for binary classification problems.
- **How it works:**
Instead of fitting a straight line, it fits an **S-shaped curve** called a **logistic function**. The output is a probability between 0 and 1, which can be used to classify data into categories (e.g., if the probability is >0.5 , classify as 1).
- **Example:**
Predicting whether a student will pass or fail based on their study hours.



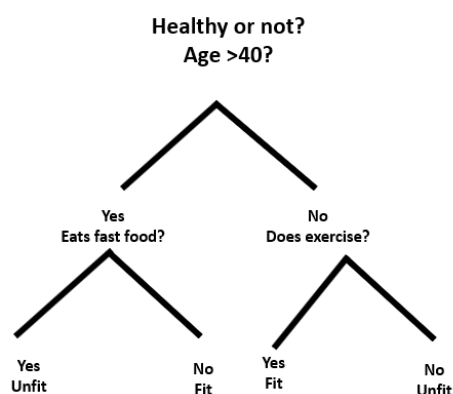
2. Support Vector Machines (SVM)

- **Definition:**
SVM is a powerful classification algorithm that works by finding the best boundary (or hyperplane) that separates data points of different classes.
- **How it works:**
SVM looks for the hyperplane that maximizes the margin between different classes. The data points that are closest to the boundary are called support vectors.
- **Example:**
Classifying emails as spam or not spam. SVM tries to find the best line (or plane) that separates the two categories.



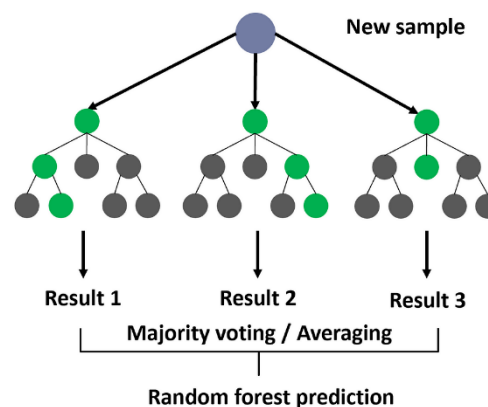
3. Decision Trees

- **Definition:**
A decision tree is a flowchart-like model where each internal node represents a decision based on a feature, each branch represents an outcome of that decision, and each leaf node represents a class label (or value for regression).
- **How it works:**
The model splits the data based on the most important features at each step, forming a tree structure. It continues to split the data into smaller and smaller subsets until a decision is made.
- **Example:**
Deciding whether to play outside based on weather conditions like "Sunny," "Windy," or "Rainy."



4. Random Forests

- **Definition:**
Random Forest is an ensemble method that combines multiple decision trees to improve the accuracy and robustness of the prediction. It creates many trees using random subsets of data and features, and the final prediction is the average (for regression) or the majority vote (for classification) of all trees.
- **How it works:**
Each decision tree gives its own prediction, and the forest takes the most common prediction (in classification) or the average prediction (in regression).
- **Example:**
Predicting whether a customer will buy a product based on their browsing history, where multiple trees analyse different aspects of the data.



5. Naive Bayes

- **Definition:**
Naive Bayes is a classification algorithm based on **Bayes' Theorem**. It assumes that all features are independent of each other, which is why it's called "naive." Despite this assumption, it works surprisingly well for many real-world applications.
- **How it works:**
Naive Bayes calculates the probability of each class given the feature values and assigns the class with the highest probability.
- **Example:**
Classifying emails as spam or not spam by calculating the likelihood of certain words (like "win," "free," "offer") appearing in spam emails.

Conditional probability: Probability of a label y occurring as a result of the given features x .

Joint probability: Probability of features x occurring together with label y .

Marginal probability: Probability of features x occurring across all possible labels y .

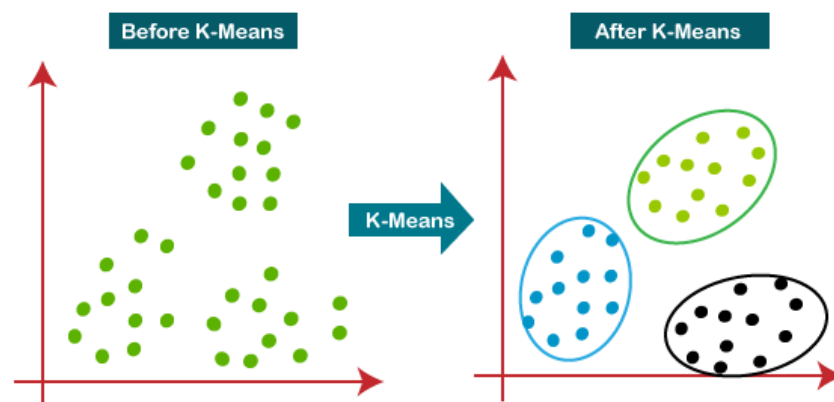
$$P(y|x) = \frac{P(x,y)}{P(x)}$$

Unsupervised learning involves finding patterns in data that isn't labelled. Here are two main types: **Clustering** (grouping similar data points together) and **Dimensionality Reduction** (simplifying data while preserving important information).

7. Clustering

1. K-Means Clustering:

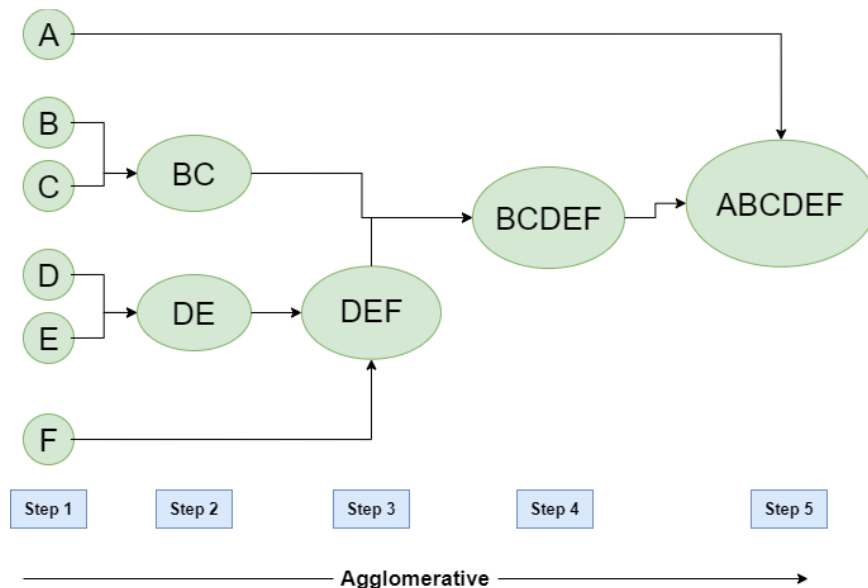
- **Definition:** K-Means is a popular clustering method that groups data into 'K' clusters based on similarity. It tries to divide the data points into K groups, where each group has data points that are close to each other.
- **How it works:**
 - First, we decide on a number, K, which is the number of clusters.
 - The algorithm randomly places K points (called centroids) in the data space.
 - Each data point is assigned to the nearest centroid.
 - The centroids are adjusted to better fit the data points assigned to them.
 - The process repeats until the centroids don't move much.
- **Example:** Grouping customers based on buying habits. One cluster could be "tech lovers," another could be "bargain hunters."



2. Hierarchical Clustering:

- **Definition:** Hierarchical clustering builds a tree (hierarchy) of clusters, either by:
 - **Agglomerative approach:** Start with each data point as its own cluster, then merge the closest clusters step by step.
 - **Divisive approach:** Start with all data points in one big cluster, and then split them step by step.

- **How it works:**
 - Each data point starts as a single cluster.
 - The algorithm repeatedly merges the two closest clusters based on their distance from each other.
 - This continues until we have the desired number of clusters or a single cluster that includes all points.
- **Example:** Building a family tree of species in biology, where related species are grouped together.



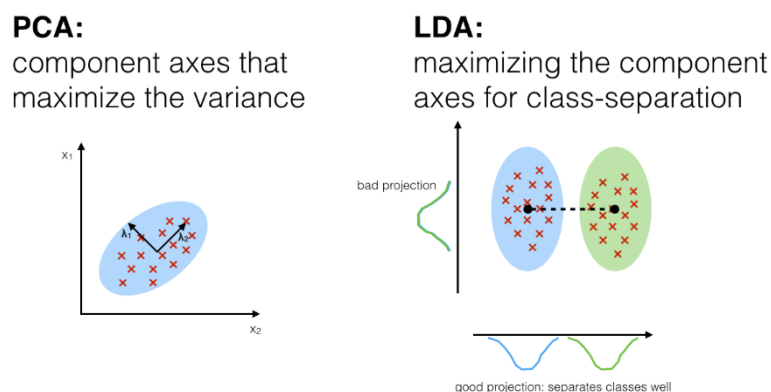
8. Dimensionality Reduction Algorithms

1. Principal Component Analysis (PCA):

- **Definition:** PCA is used to reduce the number of features (dimensions) in a dataset while keeping as much important information as possible. It transforms the data into a new set of dimensions (called principal components) that capture the most variation in the data.
- **How it works:**
 - The algorithm identifies the directions (principal components) in which the data varies the most.
 - It projects the data onto these new components, reducing the dimensions while preserving the most important information.
- **Example:** If you have a dataset with 100 features (like an image with 100 pixels), PCA might reduce it to just 10 features while keeping the important patterns.

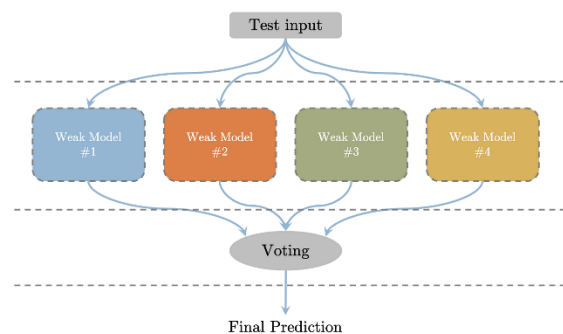
2. Linear Discriminant Analysis (LDA):

- **Definition:** LDA is also a dimensionality reduction technique, but it's mainly used when we have labelled data (making it closer to supervised learning). It finds the feature combinations that best separate the classes in the data.
- **How it works:**
 - It tries to find a new axis (or dimension) that maximizes the separation between the different classes (like distinguishing between cat and dog images).
 - The goal is to project the data onto a lower-dimensional space that keeps the differences between the classes as clear as possible.
- **Example:** LDA is used in face recognition systems, where it helps separate different faces based on key features.



9. Ensemble Learning

Ensemble learning is a technique in machine learning where multiple models (often called “weak learners”) are combined to create a stronger, more accurate model. The idea is that by combining the predictions of several models, we can reduce errors and improve performance. Here are some popular ensemble methods:



1. Bagging (Bootstrap Aggregating)

- **Definition:** Bagging is an ensemble method that involves training multiple versions of the same model on different subsets of the data and averaging their predictions. The subsets are created by randomly sampling the training data with replacement (this is called “bootstrap” sampling). By combining the predictions of many models, bagging reduces variance and helps prevent overfitting.
- **Example:** Think of it like asking multiple experts to give their opinion on a problem. By averaging their answers, you can get a more reliable prediction than from a single expert.

2. Boosting

- **Definition:** Boosting is an ensemble technique where models are trained sequentially, and each new model tries to correct the errors made by the previous ones. The idea is to improve the overall performance by focusing more on the difficult cases that earlier models struggled with.
- **How it works:** Each weak model (like a decision tree) is added one at a time, and its predictions are adjusted based on the performance of the previous models.

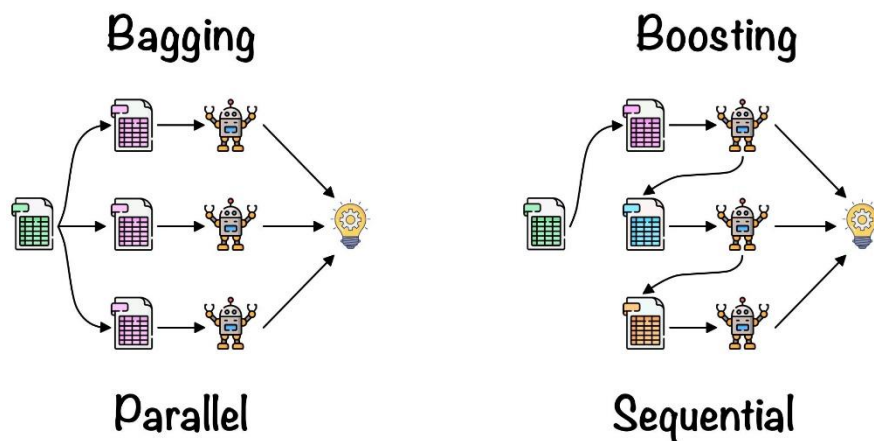
AdaBoost (Adaptive Boosting):

- **Definition:** AdaBoost works by giving more weight to data points that were incorrectly predicted by the earlier models. The model tries harder to predict these points correctly in the next iteration.
- **How it works:**
 - Initially, all data points are given equal importance.
 - After each iteration, the weights of incorrectly predicted points are increased, so the next model pays more attention to them.
 - Final predictions are made by combining the weighted predictions of all models.

Gradient Boosting:

- **Definition:** Gradient Boosting trains models sequentially like AdaBoost but focuses on minimizing the loss (or error) of the model by taking a gradient-based approach. Each new model is trained to predict the residual errors (the difference between the actual and predicted values) of the previous model.
- **How it works:**
 - The first model makes an initial prediction.
 - The next model predicts the errors (residuals) from the previous model.
 - This process is repeated, and the final prediction is the sum of all the models' predictions.

XGBoost: An efficient and scalable implementation of gradient boosting, which uses extra regularization to prevent overfitting.



10. Hyperparameter Tuning

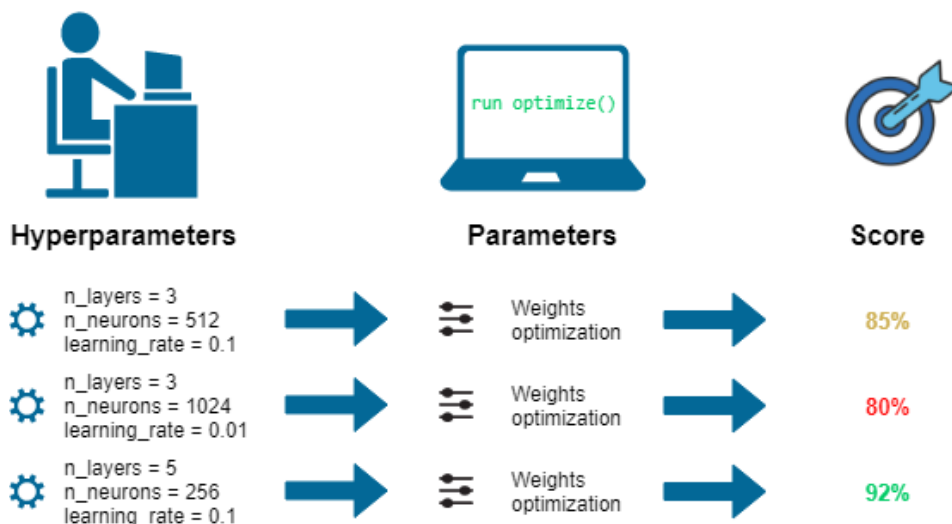
Hyperparameters are settings that are not learned by the model itself but are set by the user before training. Proper tuning of these hyperparameters can significantly improve a model's performance. Here are methods to search for the best hyperparameters:

1. Grid Search:

- **Definition:** Grid Search involves trying every possible combination of hyperparameters from a predefined set. It's an exhaustive search but can be time-consuming.
- **How it works:**
 - You define a grid of possible hyperparameter values.
 - The model is trained and evaluated for each combination of hyperparameters.
 - The combination that gives the best performance on the validation data is chosen.
- **Example:** If you're tuning a Random Forest model, Grid Search might try different values for the number of trees, max depth, etc., until it finds the best settings.

2. Random Search:

- **Definition:** Random Search is similar to Grid Search but instead of trying every combination, it selects random combinations of hyperparameters. It's faster than Grid Search but less exhaustive.
- **How it works:**
 - Instead of testing all combinations, it randomly selects a fixed number of combinations from the predefined grid and tests them.
 - The best combination is chosen based on model performance.



11. Natural Language Processing

(NLP)

Natural Language Processing (NLP) is a field of machine learning focused on the interaction between computers and human language. It enables computers to read, understand, and derive meaning from human languages in a useful way. NLP is essential for tasks like language translation, text summarization, and chatbots.

Basic NLP Techniques

a. Tokenization

- **Tokenization** is the process of breaking down text into smaller units, like words or sentences. These smaller units are called "tokens."
- **Example:** The sentence "I love machine learning" is tokenized into: "I", "love", "machine", "learning".

b. Stemming and Lemmatization

- **Stemming:** Reducing a word to its base or root form, often by chopping off the ends. Example: "Running", "runs" → "run".
- **Lemmatization:** More advanced than stemming, it reduces words to their dictionary form, considering context. Example: "Better" → "good".

c. Stop Words Removal

- **Definition:** Stop words are common words (like "the", "is", "in") that are often removed because they don't contribute much to the meaning of the text.
- **Example:** In "I am learning machine learning", words like "I" and "am" could be removed.

d. Bag of Words

- **Definition:** A method to represent text as a set of words (or tokens), ignoring grammar and word order. The frequency of each word in the text is counted and used as a feature.
- **Example:** The sentences "I love dogs" and "I love cats" would have the following Bag of Words:
"I":1,"love":1,"dogs":1,"cats":0"

e. Term Frequency-Inverse Document Frequency (TF-IDF)

- **Definition:** TF-IDF is an improvement on the Bag of Words model. It weighs words by their importance. Words that appear frequently in a document but rarely across all documents get a higher score.
- **Example:** Common words like "the" will have a low weight, while unique words like "machine" will have a higher weight in a document about machine learning.

NLP Tasks (Applications)

a. Text Classification

Example: Categorizing a movie review as positive or negative.

b. Sentiment Analysis

Example: Analysing tweets to understand public opinion on a product.

c. Machine Translation

Example: Translating an English sentence to French using tools like Google Translate.

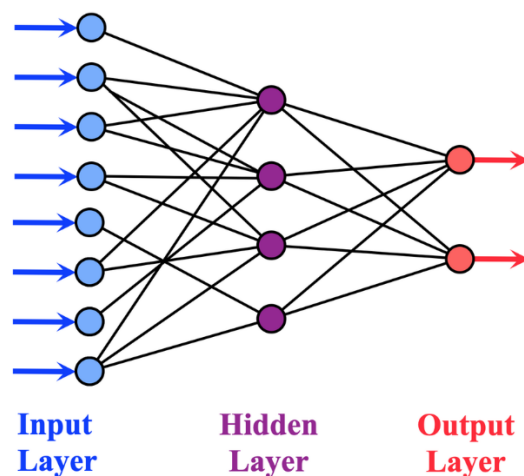
d. Text Summarization

Example: Summarizing a long news article into a few sentences.

ADVANCED ML TECHNIQUES

Deep Learning

Deep Learning is a subset of machine learning that uses neural networks with many layers to model complex patterns in data. These models learn by adjusting weights and biases through the process of backpropagation and optimization.



1. Neural Networks

A **Neural Network** is a computational model inspired by the human brain. It consists of layers of interconnected "neurons" that process input data and make predictions.

a. Artificial Neural Networks (ANN)

- **Definition:** ANNs are the basic form of neural networks where information is passed through layers of neurons. Each neuron takes input, processes it, and passes it to the next layer.
- **Structure:**
 - **Input Layer:** Takes in the data (e.g., pixels of an image, text).
 - **Hidden Layers:** Process the data through mathematical operations (like weighted sums and activation functions).
 - **Output Layer:** Produces the final prediction (e.g., a classification result).
- **Example:** In image recognition, an ANN can take an image as input and predict what's in it (e.g., cat or dog).

b. Convolutional Neural Networks (CNN)

- **Definition:** CNNs are specialized neural networks for processing grid-like data, such as images. They are designed to automatically detect patterns like edges, textures, and shapes in images.
- **Key Components:**
 - **Convolution Layers:** Use filters to scan the input data and create feature maps. Each filter learns to detect a specific feature, like a vertical edge.
 - **Pooling Layers:** Reduce the size of the feature maps by summarizing regions (e.g., taking the maximum value, called Max Pooling), which helps reduce computation.
 - **Fully Connected Layers:** Once the feature maps are processed, the CNN passes them through a regular neural network for final classification.
- **Example:** CNNs are commonly used in image recognition tasks (e.g., identifying objects in a photo) and can even be used for video analysis and medical image diagnosis.

c. Recurrent Neural Networks (RNN)

- **Definition:** RNNs are designed for sequential data, where the order of the data points matters (e.g., text, time series). RNNs maintain a "memory" of previous inputs by passing information from one step to the next.

- **Key Feature:** Each neuron has a connection not just to the next layer, but also to itself, allowing it to remember the previous input.
- **Limitations:** RNNs struggle with long sequences due to the **vanishing gradient problem**, where the network “forgets” earlier information.
- **Example:** RNNs are used for tasks like text generation, language translation, and speech recognition.

2. Transformers (latest ML model with widespread applications in recent developments)

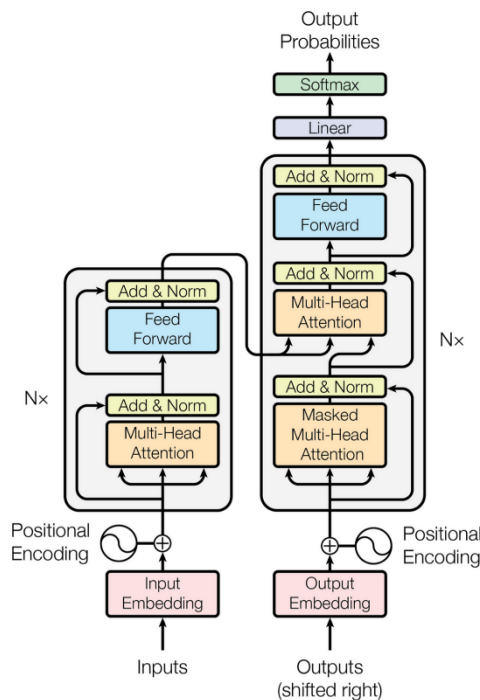


Figure 1: The Transformer - model architecture.

Transformers are state-of-the-art models that have revolutionized NLP and many other fields. Unlike RNNs, they don't process data sequentially but use a **self-attention mechanism** to capture relationships between all elements of the input, regardless of their distance in the sequence.

a. Self-Attention Mechanism

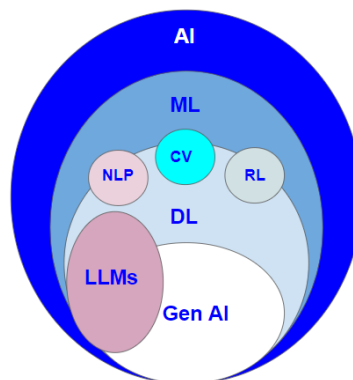
- **Definition:** Self-attention allows the model to weigh the importance of different parts of the input data when making a prediction. Each word or token in a sentence can "attend" to every other word, capturing context and meaning across the whole sequence.
- **How it works:**
 - For each word in a sentence, the self-attention mechanism computes how much focus it should give to every other word.
 - This helps the model understand relationships and dependencies in the data.

- **Example:** In a translation task, self-attention helps the model figure out which words in the source sentence relate to which words in the target sentence.

b. Encoder-Decoder Architecture

- **Definition:** The Transformer model uses an **Encoder-Decoder** structure, where:
 - **Encoder:** Processes the input data and creates a representation.
 - **Decoder:** Uses that representation to generate the output, such as translating a sentence or summarizing a text.
- **Example:** The encoder takes an English sentence, creates a deep understanding of it, and the decoder uses this understanding to generate the French translation.
- **Why it's powerful:** This architecture allows transformers to handle complex tasks like language translation, text generation, and summarization more efficiently than older models like RNNs.

Recent Developments

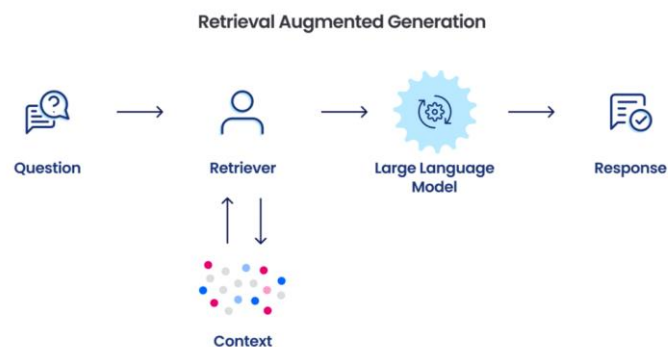


a. Large Language Models (LLMs) & Generative AI (GenAI)

- **LLMs:** are huge transformer-based models trained on massive datasets of text. They are capable of generating human-like text, answering questions, and understanding language in context.
- **Generative AI:** GenAI uses these models to generate new data, whether it's text, images, or even code. These models are fine-tuned for tasks like writing stories, creating chatbots, and generating creative content.

b. Retrieval-Augmented Generation (RAG)

- **Definition:** RAG is a hybrid model that combines the power of retrieval systems (like search engines) with generative models (like GPT). Instead of generating text purely from its training, the model retrieves relevant documents from a knowledge base and uses them to improve its answers.
- **How it works:**
 - The model retrieves relevant information from external data sources.
 - Then, it uses a generative model to provide a final, coherent answer that integrates the retrieved information.
- **Example:** A chatbot using RAG can fetch facts from a database to answer a question accurately and generate a fluent response.



c. Generative Adversarial Networks (GANs)

- **Definition:** GANs are a type of neural network used for generating realistic data, such as images, videos, or music. They consist of two models:
 - **Generator:** Tries to create realistic data.
 - **Discriminator:** Tries to distinguish between real and fake data.
- **How it works:**
 - The generator creates fake data (e.g., an image).
 - The discriminator evaluates whether the data is real or fake.
 - The two networks are trained together until the generator becomes good enough at creating realistic data that the discriminator can't tell the difference.
- **Applications:**
 - Image generation: GANs can create realistic faces of people who don't exist.
 - Data augmentation: GANs can generate new training examples to improve the performance of machine learning models.

Further Reading:

[Regression In depth](#)

[Classification in depth](#)

[Clustering in depth](#)

[Market Basket Analysis – Association Rules Application](#)

[Neural Networks](#)

[GenAI and LLMs](#)

Author:

Advait Shinde

[Check out my Blog on Machine Learning!](#)

