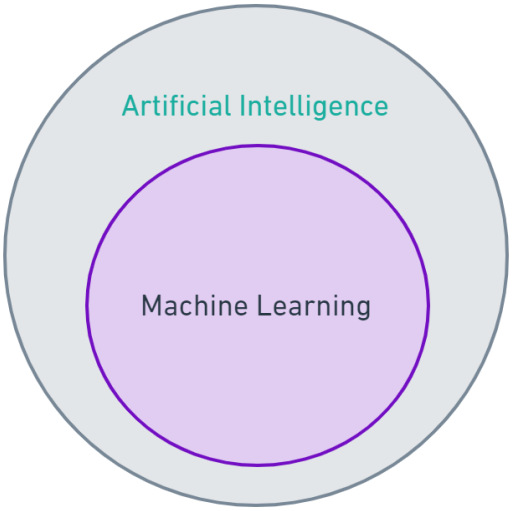
What is Machine Learning

"It is a field of study that gives the ability to the computer for self-learn without being explicitly programmed."

-Arthur Samuel

Definition:

Machine Learning is a branch of Artificial Intelligence and computer science that focuses on the ability of machines to learn. Machine learning is mainly focused on developing computer programs that can teach themselves to grow and change when exposed to new data. Machine learning studies algorithms for self-learning to do stuff. It can process massive data faster with the learning algorithm. For instance, it will be interested in learning to complete a task, make accurate predictions, or behave intelligently.



It is a system of computer algorithms that can learn from example through self-improvement without being explicitly coded by a programmer.

Machine learning is an essential component of the growing field of Data Science and Artificial Intelligence. Through statistical methods, algorithms are trained to make classifications or predictions, producing actionable insights.

In general, machine learning algorithms are used to predict or classify. Based on some input data, which can be labeled or unlabeled, your algorithm will estimate a pattern in the data.

A typical machine learning tasks are to provide a recommendation. For those who have an Instagram account, all advertisements are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience by personalizing recommendations.
Machine learning is also used for various tasks like fraud detection, predictive maintenance, portfolio optimization, automatization task, etc.

Why is Machine Learning Needed?

- Machine learning is growing in importance due to increasingly enormous volumes and variety of data, the access and affordability of computational power, and the availability of high-speed Internet.

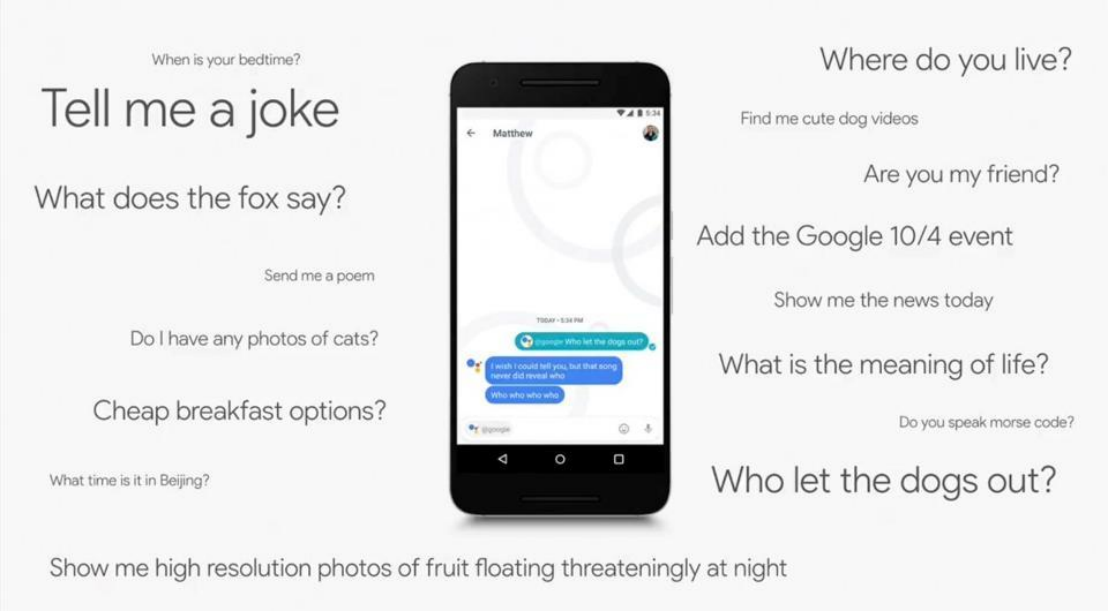
- Data is growing day by day. It is impossible to understand all data with higher speed and higher accuracy. More than 80% of the data is unstructured: audios, videos, photos, documents, graphs, etc. The information has been very massive. The time taken to compute would increase, where Machine Learning comes into action, to help people with important data in minimum time.

- And now, machine learning is present in so many technology segments that we don't even realise it while using it.

Real World Example of Machine Learning

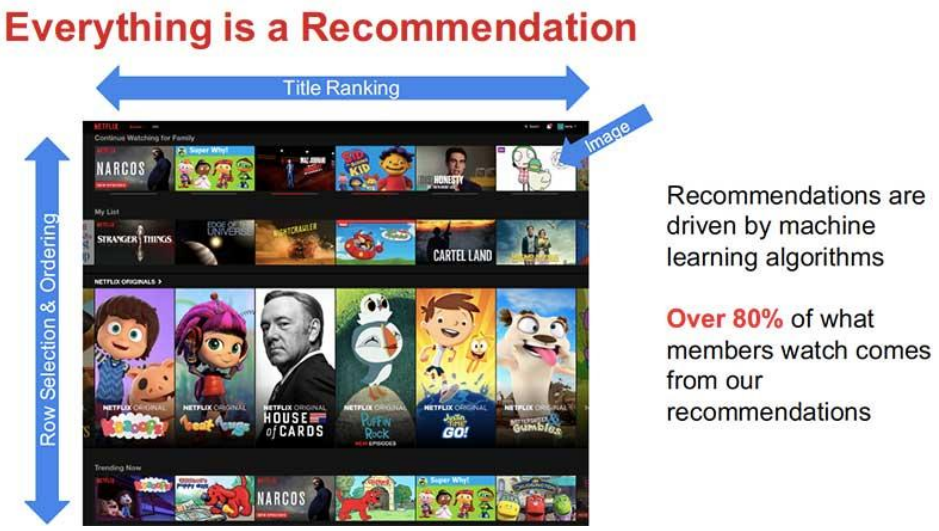
❖ Google Assistant / Siri:-

A personal digital assistant and one of the world's biggest Machine Learning projects have ever seen. Google Assistant can interact with your Android phone to do various tasks, such as setting alarms or playing music. it can even handle some home automation devices.
It provides a virtual personal assistant experience through a natural language speech interface to perform a variety of tasks.



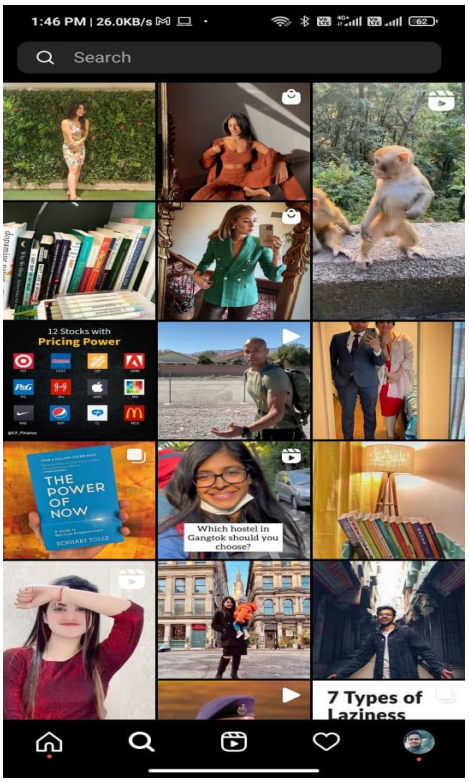
❖ Netflix:-

Netflix uses an ML technology called a “recommendation engine” to suggest shows and movies to you and other users. Netflix can understand which films to recommend to other "similar" users by analyzing the ratings. Machine learning allows the platform to automate millions of decisions based on user activities.



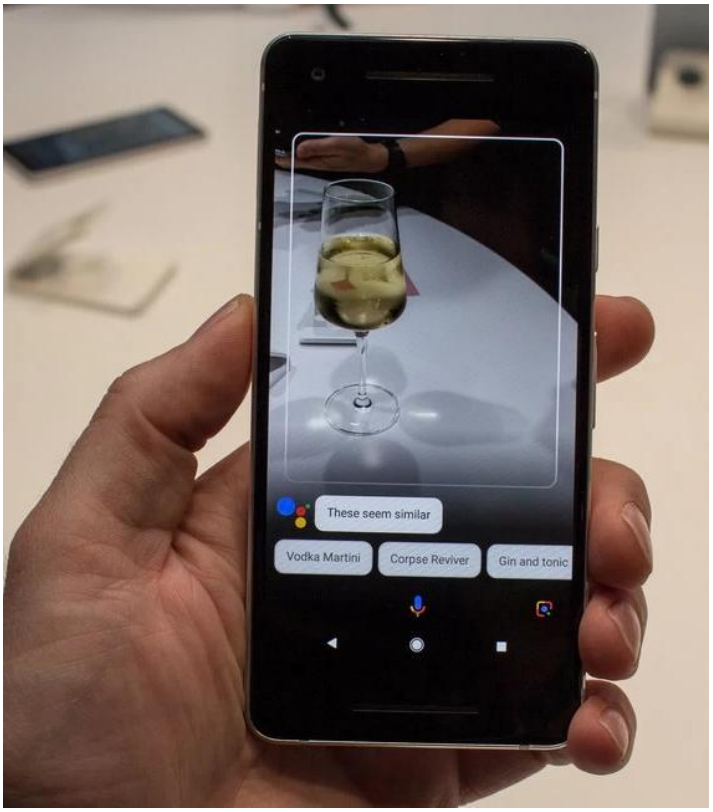
❖ Instagram:-

Instagram uses machine learning to prioritize posts based on relevance, freshness, popularity, and user preference. Instagram displays a range of posts on a user's home feed and explore list, ranging from the most popular posts to the most followed accounts.



❖ Google Lens:-

It is an AI-powered technology that uses your smartphone camera and deep machine learning to not only detect an object in front of the camera lens but understand it and offer actions such as scanning, translation, shopping, and more. Google Lens enables you to point your phone at something, such as a specific flower, and then ask Google Assistant what the object you're pointing at is. You'll not only be told the answer, but you'll get suggestions based on the thing, like nearby florists, in the case of a flower.



❖ Medical Diagnosis:-

Machine learning can be used in the techniques and tools that can help diagnose diseases. It is used to analyze the clinical parameters and their combination for the prognosis example, prediction of disease progression for the extraction of medical knowledge for the outcome research, therapy planning, and patient monitoring.

❖ Financial Services:-

Machine learning can help banks and financial institutions to make smarter decisions. Machine learning can help financial services to spot an account closure before it occurs. It can also track the spending pattern of the customers. Machine learning can also perform market analysis. Smart machines can be trained to track spending patterns. The algorithms can identify the trends easily and can react in real-time.

Important Terminologies

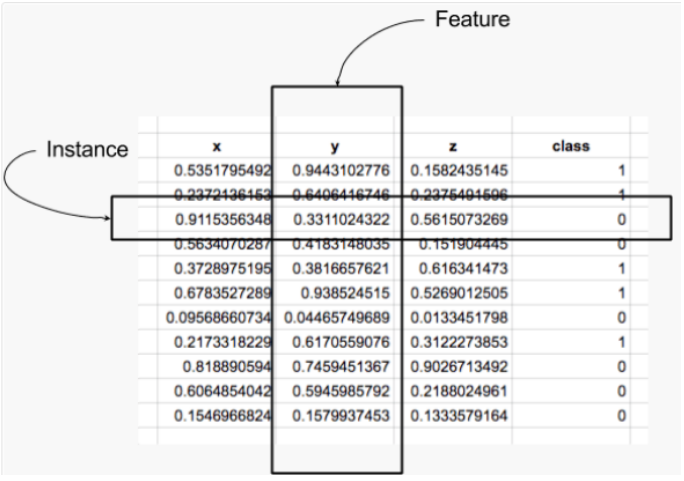
- **Instance:** A single row of data is called an instance. It is an observation from the domain.

- **Feature:** A single column of data is called a feature. It is individual independent variables that act as input in your system. While making the predictions, models use such features to make the predictions. It is a measurable property that can predict the output of a given dataset. It may be possible that the output doesn't depend on a particular feature, but all features initially equally contribute to the output.

- **Dataset:** A data set is a collection of instances.

- **Training Dataset:** A dataset that we feed into our machine learning algorithm to train our model.

- **Testing Dataset:** A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset.



- **Model:** A machine learning model is a file that has been trained to recognize certain types of patterns. It is the output of a machine learning algorithm run on data. A model represents what was learned by a machine learning algorithm.

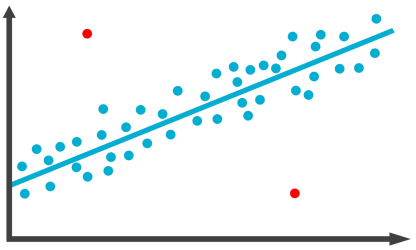
- **Training a model:** Training a model refers to feeding data to a learning algorithm and allowing it to run over the fed data to learn or find the patterns over time to predict expected results.

- **Target/Label:** A label is a thing we're predicting or which can be the output. For example, if a model predicts that the input image is cat or dog, here cat and dog are label.

- **Overfitting** happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the model's performance on new data. For example, if our model saw 99% accuracy on the training set but only 55% accuracy on the test set.

- **Underfitting:** It is opposite to what underfitting means. It happens when the model cannot accurately capture the relationship between the input and output variables, generating a high error rate on both the training set and unseen data.

- **Outlier:** Outliers are extreme values that fall far beyond the other observations. For example, if a feature has data in the range 200-500, and it has one data which is 10,000, then this data(10,000) is known as an outlier.



In this image, blue dots are the general dataset points are plotted in a graph and red dots are outliers.

Prerequisites before starting with the Track

Machine Learning Track does not presume or require any prior knowledge in machine learning. However, to understand the concepts presented, we recommend that learners meet the following prerequisites:

1. **Linear Algebra**:- Linear algebra deals with vectors, matrices, and linear transformations. It is essential in machine learning as it can be used to transform and perform operations on the dataset.
    a. variables, coefficients, and functions
    b. linear equations such as $y=b+w_1x_1+w_2x_2$
    c. logarithms, and logarithmic equations such as $y=\ln(1+ez)$
    d. sigmoid function
    e. matrix multiplication

2. **Probability**:- Probability helps predict the likelihood of the occurrences. It helps us reason the situation may or may not happen again. For machine learning, the probability is a foundation.
    a. Notation
    b. Probability distribution, joint and conditional
    c. Different rules of probability—Bayes theorem, sum rule, and product or chain rule
    d. Independence
    e. Continuous random variables

3. **Statistics**:- Statistics contain tools that can be used to get some outcome from the data.
    a. Mean
    b. Median
    c. Standard deviation
    d. Outliers
    e. Histogram

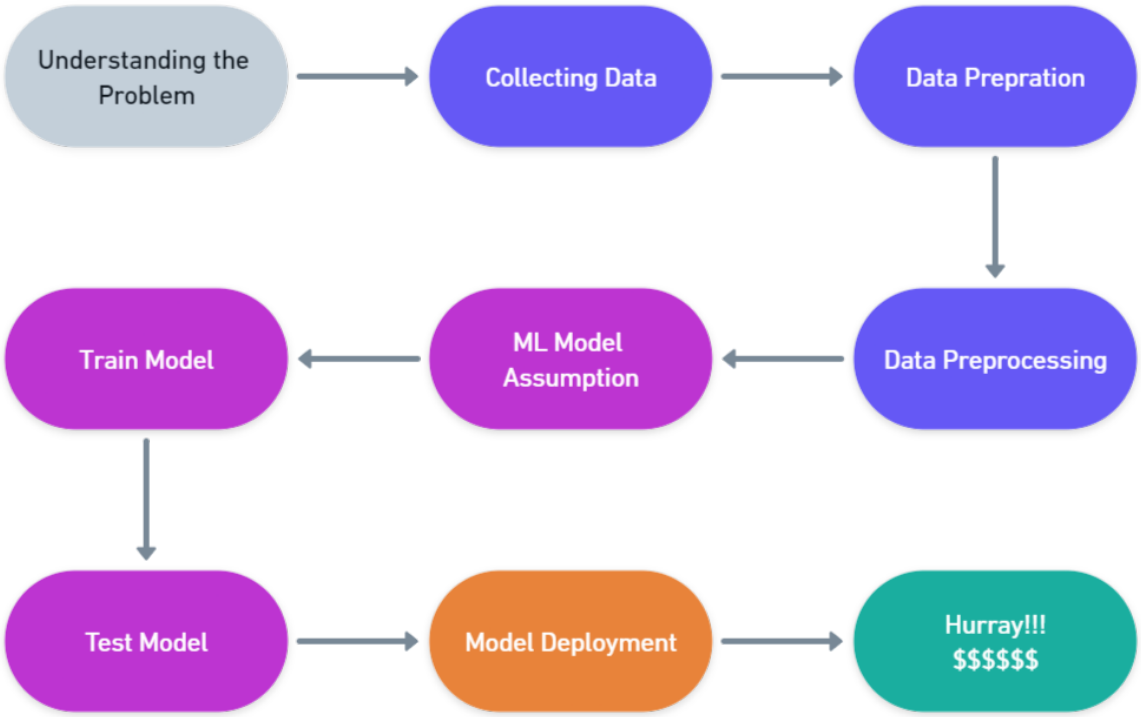
4. **Calculus**:- Calculus is an important field in mathematics, and it plays an integral role in many machine learning algorithms.
    a. Basic knowledge of integration and differentiation
    b. Partial derivatives
    c. Gradient or slope
    d. Chain rule—for training neural networks

5. **Programming Language**:- It is essential to know programming languages like Python or R to implement the whole Machine Learning process. Python and R both provide in-built libraries that make it very easy to implement Machine Learning algorithms. In this track, we will be using python.
    a. Lists, sets, and dictionaries (assessing, iterating, and creating)
    b. for loops with multiple variable iterators
    c. if/else conditional expressions
    d. String formatting
    e. Pass statement – for syntax
    f. Defining and calling functions
    g. Numpy and Pandas library will be a plus point.

You do not require in-depth knowledge of each topic, and just basic knowledge is okay.

Life Cycle of Machine Learning

We divide the Machine Learning Life Cycle into eight major steps for easy understanding of the Machine Learning Life Cycle.



1. **Understanding the Problem:** Understand the problem and the use case you are working with and define a proper problem statement. By defining problems properly, you make them easier to solve, which means saving time, money, and resources.

2. **Collecting Data:** The next step is to collect and prepare all relevant data for machine learning. The more data will be, the more accurate the prediction will be. Collect as much raw data as possible regardless of quality, in the end, and it will be taken care of in the following steps. It is

helpful to have a lot of data available to add as needed when problems arise with model performance.

3. **Data Preparation:** After collecting the data, we need to prepare it for further steps. The data need to be merged from different databases when dealing with larger datasets or various sources. Understanding the raw data and preparing the data in the required format is one of the significant parts of data preparation.

4. **Data Preprocessing:** It is the process of cleaning and converting raw data into a usable format and making it suitable for a machine learning model. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues. The data may contain duplicate entries, which need to be removed in most applications. Based on the problem, a missing value replacement needs to be chosen. Splitting the data into train and test data is also part of Data Preprocessing.

5. **ML model Assumption and Check:** Various ML models are built based on the problem statement. It starts with the determination of the type of problems. We select machine learning techniques such as Classification, Regression, Cluster analysis, Association, etc., then build the model using prepared data and evaluate the model.

   Hence, we take the data and use machine learning algorithms to build the model in this step.

6. **Train Model:** Now, the next step is to train the model. In this step, we train our model to improve its performance for a better outcome of the problem. We use datasets to train the model using various machine learning algorithms. Training a model requires understanding the multiple patterns, rules, and features.

7. **Test Model:** Once you manage to get a model that has learned your training data, it's time to dig in and see how well it can perform on new data. In this step, we check for the accuracy of our model by providing a test dataset to it. Predict and review the outputs after fine-tuning the model.

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

What is Data in Machine Learning

What is Data?

Data can be defined as a representation of facts, concepts, or instructions in a formalized manner, which should be suitable for communication, interpretation, or processing by a human or electronic machine. Data can exist in various forms: as numbers or text recorded on paper, as bits or bytes stored in electronic memory, or as facts living in a person's mind.
It can be any unprocessed fact, value, text, sound, or picture that is not being interpreted and analyzed.

Raw data: Information that has been collected but not formatted or analyzed.

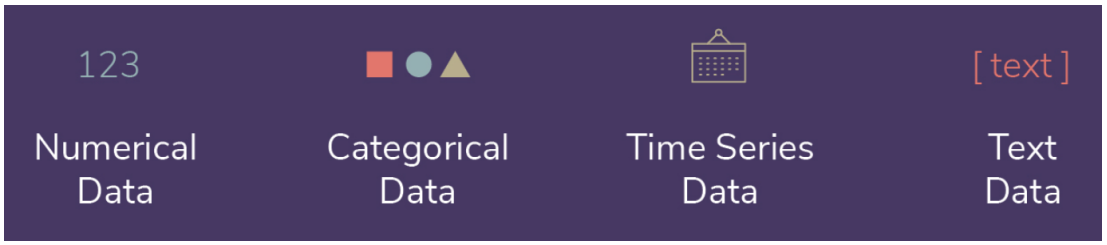
Information is organized or classified data, which has some meaningful values for the receiver. Information is the processed data on which decisions and actions are based.

Data may be collected either in structured form or unstructured form. Unstructured data is very difficult to deal with. We can't make much inference with this. So the data needs to be in a structured form. The Structured form of data used to gather the inferences is called Information.

Data is the essential part of all Data Analytics, Data Science, Machine Learning, Artificial Intelligence. We can't train any model without data, and all modern research and automation will go in vain. Big Enterprises are spending lots of money to gather as much specific data as possible.

What type of data does machine learning need?

Data can come in many forms, but machine learning models rely on four primary data types. These include numerical, categorical, time series, and text data.



- **Numerical Data:** Numerical data, or quantitative data, is any form of measurable data such as your height, weight, or the cost of your car. Exact or whole numbers (i.e., 26 students in a class) are considered discrete numbers. In contrast, those which fall into a given range are considered continuous numbers. They are simply raw numbers. You can do mathematical operations in such data.

- **Categorical Data:** Categorical data is a collection of information that is divided into groups. This can include gender (ex:- male or female), social class, ethnicity, hometown, the industry you work (ex:- Fintech or Edtech) in, or a variety of other labels. It is non-numerical, meaning you are unable to do any mathematical operations.

- **Time Series Data:** Time series data consists of data points indexed at specific time points. Learning and utilizing time series data makes it easy to compare data from week to week, month to month, year to year, or according to any other time-based metric you desire.

  The distinct difference between time series and numerical data is that time-series data have established starting and ending points. In contrast, numerical data simply collects numbers that aren't rooted in particular time periods.

- **Text Data:** Text data is simply words, sentences, or paragraphs that can provide some level of insight to your machine learning models. Since these words can be difficult for models to interpret on their own, they are

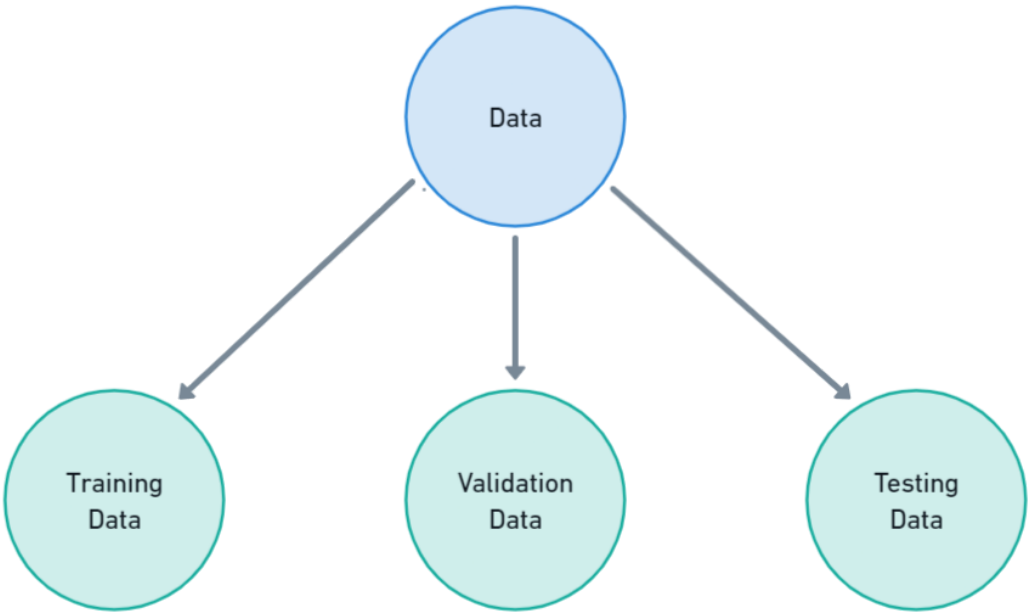
most often grouped together or analyzed using various methods such as word frequency, text classification, or sentiment analysis.

What is a Dataset?

A single row of data is called an **instance.**

Datasets are a collection of instances that all share a common attribute.
For machine learning models to understand how to perform various actions, training datasets must first be fed into the machine learning algorithm, followed by validation datasets (or testing datasets) to ensure that the model is interpreting this data accurately.
Once you feed these training and validation sets into the system, subsequent datasets can then be used to sculpt your machine learning model going forward. The more data you provide to the ML system, the faster that model can learn and improve.

How do we split data in machine learning?



- **Training Data:** A data that we feed into our machine learning algorithm to train our model. This is the data that your model actually sees(both input and output) and learns from.

- **Validation Data:** The part of data used to evaluate the model frequently, fit on the training dataset and improve involved hyperparameters (initially set parameters before the model begins learning). This data plays its part when the model is actually training.

- **Testing Data:** A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset. Once our model is completely trained, testing data provide an unbiased evaluation. When we feed in the inputs of Testing data, our model will predict some values (without seeing actual output). After prediction, we evaluate our model by comparing it with the actual output present in the testing data. This is how we evaluate and see how much our model has learned from the experiences feed in as training data set at the training time.

How to get Datasets

To get the best result for our Machine Learning algorithm, we need a lot of data to train our model. But fetching this large amount of data manually is nearly impossible. Collecting and preparing the dataset is one of the most crucial parts while creating an ML/AI project.

Some popular sources of Machine Learning datasets for learning:-

- **Kaggle** is one of the best platforms for learning Data Science and Machine Learning. Almost everyone in his learning journey goes through this site at least once. Kaggle provides a high-quality dataset in different formats that we can easily find and download.

- **Awesome Public Datasets** in Github. **GitHub** is the world standard for collaborative and open-source code repositories online, and many projects it hosts have datasets you can use. Awesome Public Datasets in Github provide high-quality datasets. Datasets are divided and arranged in a well-organized manner within a list according to topics such as Agriculture, Biology, Climate, Complex networks, etc. Most of the data sets listed below are free; however, some are not.

- **Scikit Learn Datasets** provides clean datasets (i.e., ready to be used to train an ML model) for you to use when building ML models. The datasets come with the Scikit-Learn package itself. You don't need to download anything. Within just a few lines of code, you'll be working with the data. Scikit-Learn provides seven datasets. These datasets are robust and serve as a strong starting point for learning ML.

- The **UCI Machine Learning Repository** is a collection of databases, domain theories, and data generators used by the machine learning community for the empirical analysis of machine learning algorithms. UCI Machine Learning Repository is a collection of over 550 datasets.

- **Government Datasets**, Various countries publish government data for public use from different departments. The goal of providing these datasets is to increase the transparency of government work among the people and use the data in an innovative approach. There are different sources to get government-related data.

- **Google Datasets Search Engine** is a search engine for data sets that helps researchers locate online data that is freely available for use. It has more than 25 million datasets available. It has a ton of filters to narrow down results.

Myths about Machine Learning

- **Myth:** Artificial Intelligence and Machine Learning are same thing.
  Reality: Machine Learning is a subset of Artificial Intelligence.

- **Myth:** Machine Learning is all about predicting the future.
  Reality: Machine Learning is good at predicting the future when the future looks like the past.

- **Myth:** Machine Learning is about computers learning to think like humans.
  Reality: Machine Learning is about "learning" patterns in data.

- **Myth:** Machine Learning is about delivering higher accuracy.
  Reality: Accuracy is never a good measure of model performance.

- **Myth:** Machine Learning automatically gets better over time.
  Reality: Once deployed, ML models become less accurate until they are retrained.

- **Myth:** Machine Learning is a black box.
  Reality: Machine Learning is just harder to explain, but it is far more transparent than people.

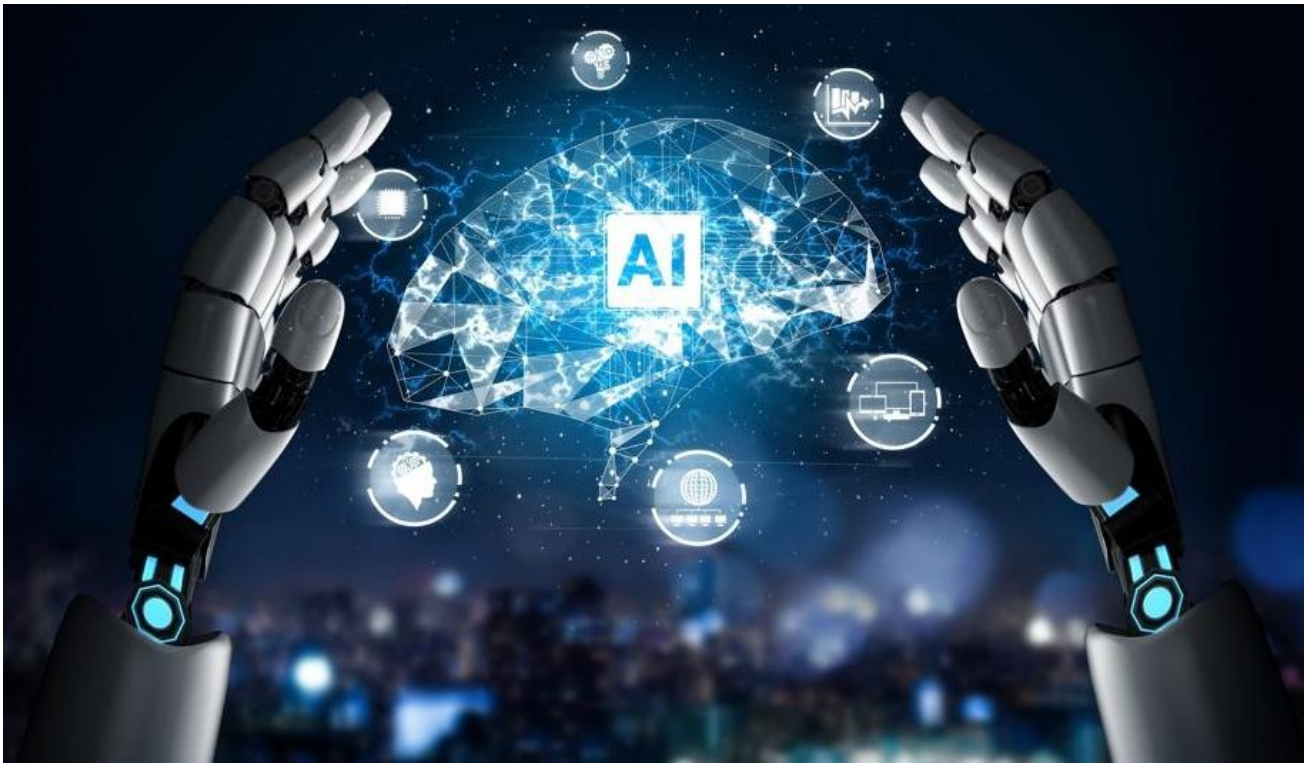
- **Myth:** Machine Learning can be used anywhere.
  Reality: It is not worth to use machine learning for small data solutions as that can be done by a human effortlessly.

- **Myth:** Machine Learning will take over Human Work.
  Reality: Machine learning will create more significant opportunities for new skills and creative thinking.

Introduction to Artificial Intelligence

Definition:

Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building intelligent machines capable of performing tasks that typically require human intelligence.



Artificial intelligence makes it possible for machines to learn from experience, adjust to new inputs and perform human-like tasks. Most AI examples that you hear about today – from chess-playing computers to self-driving cars – rely heavily on deep learning and natural language processing. Using these technologies, computers can be trained to accomplish specific tasks by processing large amounts of data and recognizing patterns in the data.

AI gets the most out of data. When algorithms are self-learning, the data itself is an asset. AI works by combining large amounts of data with fast, iterative processing and intelligent algorithms, allowing the software to learn automatically from patterns or features in the data.

Every industry has a high demand for AI capabilities – including systems that can be used for automation, learning, legal assistance, risk notification, and research.

Importance of Artificial Intelligence:-

- AI automates repetitive learning and discovery through data.
- AI adds intelligence to existing products.
- AI adapts through progressive learning algorithms to let the data do the programming.
- AI analyzes more and deeper data using neural networks with many hidden layers.
- AI achieves incredible accuracy through deep neural networks.

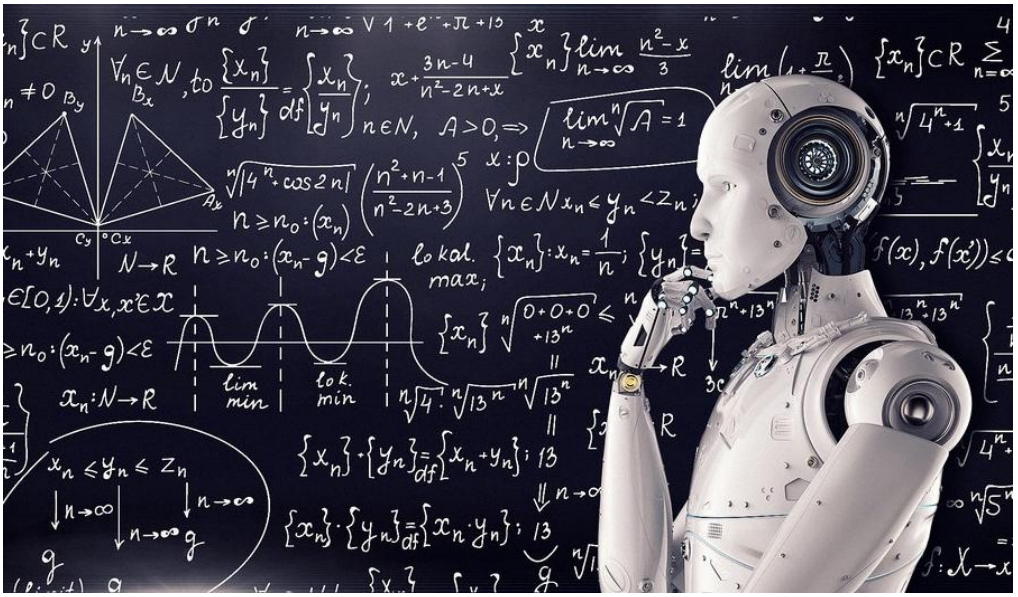
Examples of Artificial Intelligence:

- Siri, Alexa, Google Assistant, and other smart assistants,
- Self-driving cars,
- Image recognition software,
- Chatbots or Conversational bots,
- Recommendation engines can provide automated recommendations for TV shows based on users' viewing habits, like Netflix's recommendations,.
- Robo-advisors,
- Email spam filters

Weak Vs Strong AI:

Weak AI, also known as Narrow AI, is an AI system that is designed and trained to complete a specific task. Industrial robots and virtual personal assistants, such as Apple's Siri, use weak AI. Weak AI drives most of the AI that surrounds us today.

Strong AI, also known as artificial general intelligence (AGI), describes programming that can replicate the cognitive abilities of the human brain. When presented with an unfamiliar task, a robust AI system can use fuzzy logic to apply knowledge from one domain to another and find a solution autonomously. Artificial general intelligence (AGI), or general AI, is a theoretical form of AI where a machine would have an intelligence equal to humans; it would have a self-aware consciousness that can solve problems, learn, and plan for the future. Example: Advanced Robotics.

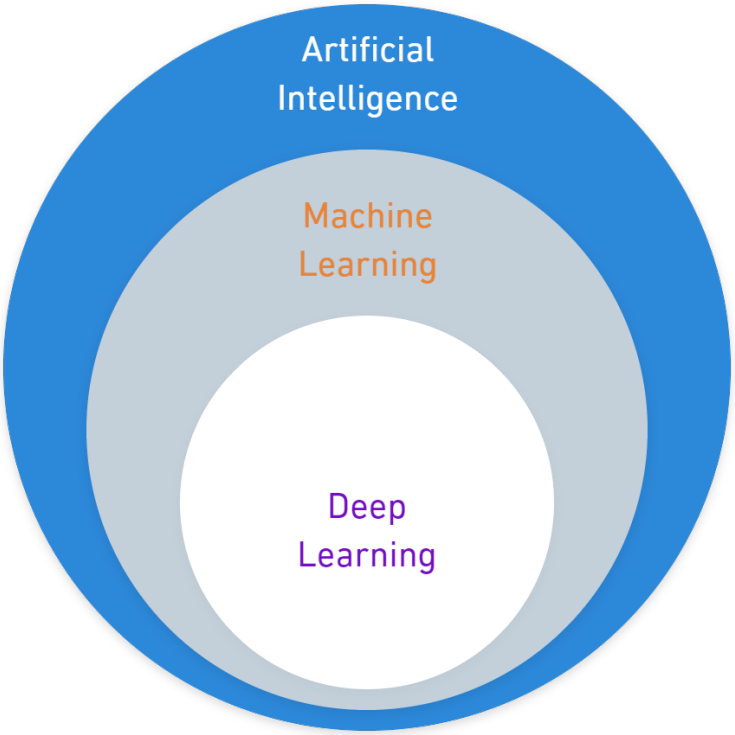


AI VS ML

Artificial intelligence and machine learning are very closely related and connected. Even though both AI and ML are based on statistics and mathematics, they are not the same thing.

Artificial Intelligence	Machine Learning
AI enables machines to think without any human intervention. It is a broad area of computer science. AI is a technique that enables machines to mimic human behavior.	ML is a subset of AI that uses statistical learning algorithms to build intelligent systems. The ML systems can automatically learn and improve without explicitly being programmed.
The focus is on improving learning capabilities along with thinking capabilities.	The focus is on improving learning capabilities.
AI is decision-making.	ML allows systems to learn new things from data.
AI systems are proficient in learning, reasoning, and self-correction.	ML systems are proficient in learning and self-correction when provided with new data.
The main objective of artificial intelligence is to maximize the chances of success and not accuracy.	The main aim of machine learning is to maximize the accuracy of predicted output and not the success ratio.
Machine learning and deep learning are the two main subsets of AI.	Deep learning is the main subset of machine learning.
In AI, we make intelligent systems to perform tasks like humans.	In ML, we teach machines with data to perform a particular task and give an accurate result.

Machine learning and deep learning are the two main subsets of AI.	Deep learning is the main subset of machine learning.
Good examples of AI are Apple Siri, Google Assistant, Tesla self-driving cars, Amazon Alexa, etc.	Good examples of machine learning are Google search engines, Twitter sentiment analysis, stock prediction, news classification, etc.



To sum things up, AI solves tasks that require human intelligence, while ML is a subset of artificial intelligence that solves specific tasks by learning from data and making predictions.

This means that all machine learning is AI, but not all AI is machine learning. All machine learning falls under the AI umbrella. But artificial intelligence is much more than only machine learning.

Anaconda

What is Anaconda?

Anaconda is the same as any other virtual environment (collection of tools or packages), but it offers more than managing environments. Anaconda has its python package manager called conda, and at the same time, Anaconda supports other package managers like pip,pipenv, and others. Conda is a package manager who takes care of different packages by handling, updating, and removing them. Anaconda supports other languages like R too.

Anaconda comes with over 150 installed data science packages, everything we can imagine.

A **package** is a code someone else has written that can be used for specific purposes. We can use a package at our convenience. Packages are helpful because you would have to write more code to get what you need to do without them.

Anaconda is a hardware store for data scientists with every tool from training datasets to modeling them. In short, Anaconda is all in one package.

Why do we use Anaconda?

- **No directory needed:** Unlike other virtual environments like virtualenv, we do not specify the directory before setting up an environment. Anaconda enables us to start a new virtual environment without worrying about its path.

- **Python Version:** As long as the python version exists in the server, Anaconda's conda can easily create the environment by grabbing the exact version of Python from the server.

- **Conda Package Manager:** Instead of just managing different packages like another package manager, conda does more than that by updating and removing packages.

Anaconda provides the ability to share the foundation of our work. Anaconda ensures that if someone else reproduces our work, they have the same tool as us.

Suppose we are working on a dataset where we should have tools to explore the data, train our model and visualize graphs for better understanding; this is where Anaconda comes to the picture. Anaconda provides all these essential tools in one place, eventually making our tasks more manageable.

What makes Conda environment special from other virtual environments?

A Virtual environment is used to manage python packages for different projects; however, there are few differences between Conda and Virtual environments.

- Other virtual elements are not virtual agnostic,i.e., different virtual environments are Python-specific, whereas conda environments are not.
- Virtual environments depend on the base system install of Python, whereas Conda environments are independent of the base install.
- Conda packages are binary, whereas libraries in Virtual environments are packaged as source distributions.

Anaconda Installation

Now, let's look at the installation of Anaconda in our system.

Reference: Anaconda Official Website

Installation on Windows:

- Download Anaconda Installer (windows version) from the below link - [Note: Install 3+ version] https://www.anaconda.com/products/individual#Downloads
- Locate your download and double-click it.
- Click Next.
- Read the licensing terms and click “I Agree.”
- Select an install for “Just Me” unless you’re installing for all users (which requires Windows Administrator privileges) and click Next.
- Select a destination folder to install Anaconda and click the Next button.
- Choose whether to add Anaconda to your PATH environment variable. We recommend not adding Anaconda to the PATH environment variable since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.
- Click the Install button. If you want to watch the packages Anaconda is installing, click Show Details.
- Click the Next button.

That is all from the installation path. Now, you can launch a jupyter notebook, and it will redirect us to our web browser and create a new notebook.
We do not have to install the jupyter notebook separately. It is already present inside Anaconda.

Installation on macOS:

- Download Anaconda(macOS version) from the below link - [Note: Install 3+ version] https://www.anaconda.com/products/individual#Downloads

- Locate your download and double-click it.
- Follow the prompts on the installer screens.
- You'll be prompted to give your password, which is usually the one that you also use to unlock your Mac when you start it up. After entering your password, click on Install Software
- You should get a screen saying the installation has been completed. Close the installer and move it to the trash.
- To make the changes take effect, close and then re-open your Terminal window.

Installing on Linux:

- Download Anaconda(Linux Version) from the below link - [Note: Install 3+ version] https://www.anaconda.com/products/individual#Downloads
- In your terminal window, enter the following to install Anaconda for Python 3.7: -
  bash ~/Downloads/Anaconda3-2020.02-Linux-x86_64.sh
- Follow the instruction on the terminal window
- You'll be prompted to choose the installation location. You can press ENTER to accept the default location.
- When the installation is finished, close and then re-open your Terminal window to make the changes take effect.

How to check Anaconda is Installed or not:

- In your Terminal window or Anaconda Prompt, run the command: conda --version.
- It will show the installed version of the anaconda. Else, it shows the error (command not found).

How to update Anaconda from the older version:

You can easily update Anaconda to the latest version

Windows:

- Open the Start Menu and choose Anaconda Prompt

- Enter these commands:

```
conda update conda
conda update anaconda
```

macOS or Linux:

- Open a terminal window
- Enter these commands:

```
conda update conda
conda update anaconda
```

Anaconda Navigator and Python Program

Reference: Anaconda Documentation

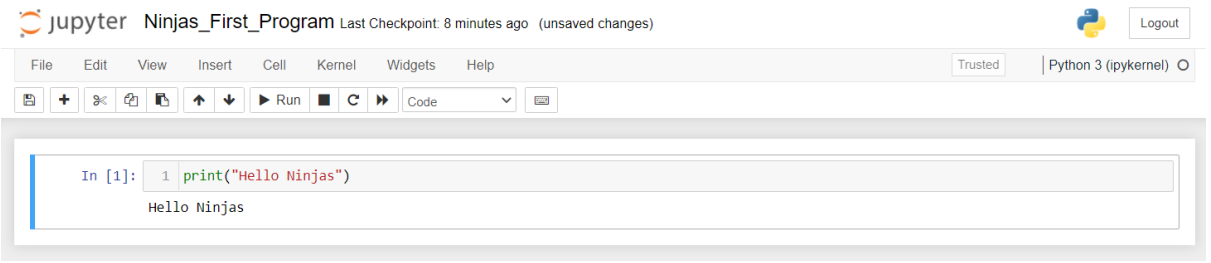
Navigator is a desktop graphical user interface that allows you to launch applications and efficiently manage conda packages, environments, and channels without using command-line commands.

Writing First Python Program:-

Use Anaconda Navigator to launch an application. Then, create and run a simple Python program with Jupyter Notebook.

- Open Anaconda Navigator
  - In **Windows** from the Start menu, click the Anaconda Navigator desktop app.
  - In **macOS**, open Launchpad, then click the Anaconda Navigator icon.
  - In **Linux**, open a terminal window and type anaconda-navigator.
- On Navigator's Home tab, in the Applications pane on the right, scroll to the Jupyter Notebook tile and click the Install button to install Jupyter Notebook.
- Launch Jupyter Notebook by clicking Jupyter Notebook's Launch button. This will launch a new browser window (or a new tab) showing the Notebook Dashboard.
- On the top of the right-hand side, a dropdown menu is labeled "New." Create a new Notebook with the Python version you installed.

- Rename your Notebook. Click on the current name and edit it or find rename under File in the top menu bar. You can name it to whatever you'd like, but we'll use Ninjas_First_Program for this example.
- In the first line of the Notebook, type or copy/paste print("Hello Ninjas").
- Save your Notebook by either clicking the save and checkpoint icon or selecting File - Save and Checkpoint in the top menu.
- Run your new program by clicking the Run button or selecting Cell - Run All from the top menu.

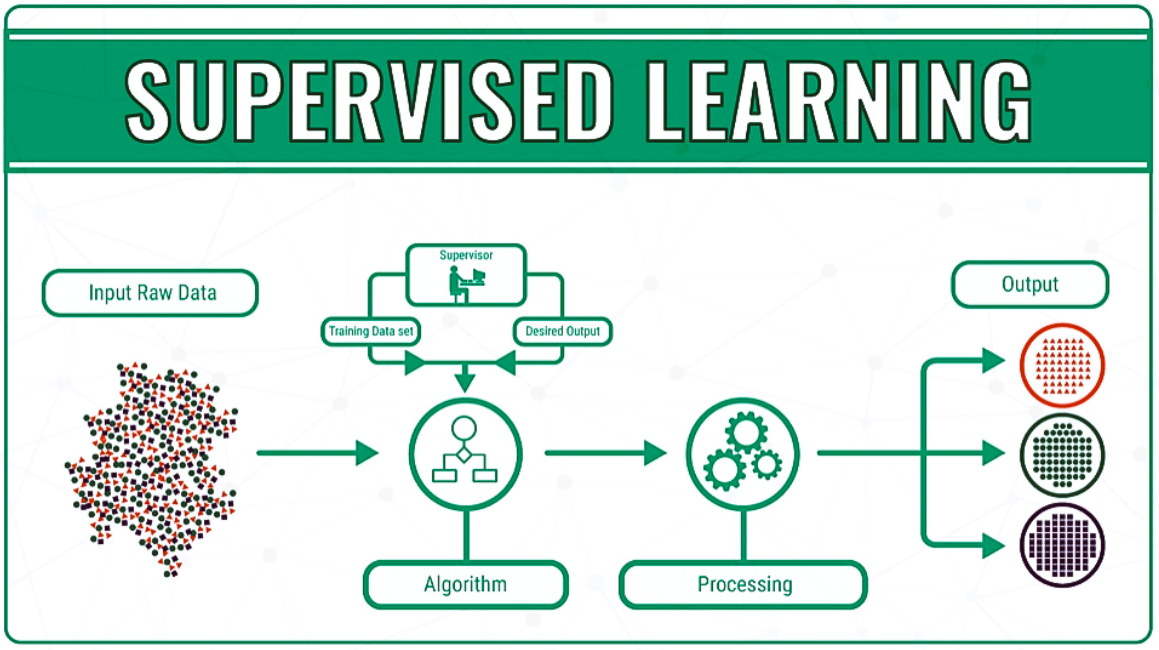


Supervised Learning

Definition

Supervised machine learning algorithms are designed to learn by example. When training a supervised learning algorithm, the training data will consist of inputs paired with the correct outputs.

Supervised learning is machine learning in which machines are trained using well "labeled" training data, and based on that data; machines predict the output. The **labeled data** means input data is already tagged with the correct output. Its algorithms are characterised by using labeled datasets to train algorithms that properly segregate data or predict outcomes. As input data is fed into the model, the weights are adjusted until the model is well fitted, which occurs as part of the cross-validation process.



The model is trained until it can detect the underlying patterns and relationships between the input data and the output labels, enabling it to yield accurate labeling results when presented with never-before-seen data.

Supervised learning helps organizations solve a variety of real-world problems at scale, such as:-

- **Spam detection**
  Companies can train databases to spot patterns or abnormalities in fresh data using supervised classification algorithms to categorize spam and non-spam email exchanges efficiently.
- **Image/Object recognition**
  Supervised learning methods may be used to find, isolate, and categorize objects in movies or pictures, making them valuable in computer vision techniques and visual analysis.
- **Predictive analytics**
  Predictive analytics enables organisations to forecast certain outcomes depending on a particular output variable, assisting business executives in justifying actions or pivoting for the benefit of the firm.

Supervised learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimised.

Supervised learning is further divided into two types:-

- **Regression** is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business.
  A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

- **Classification** uses an algorithm to assign test data into specific categories accurately. It recognizes particular entities within the dataset

and attempts to draw some conclusions on how those entities should be labelled or defined. A classification problem is when the output variable is a category, such as “Red” or “blue” or “disease” and “no disease”.

Advantages of Supervised Learning:

- With the help of supervised learning, the model can predict the output based on prior experiences.
- A supervised learning model helps us to solve various real-world problems such as fraud detection, spam filtering, etc.
- You can specifically determine how many classes you want to have.

Disadvantages of Supervised Learning:

- Training the model required lots of computation times.
- Pre-processing of data is one of the biggest challenges
- Supervised learning models are not suitable for handling complex tasks.

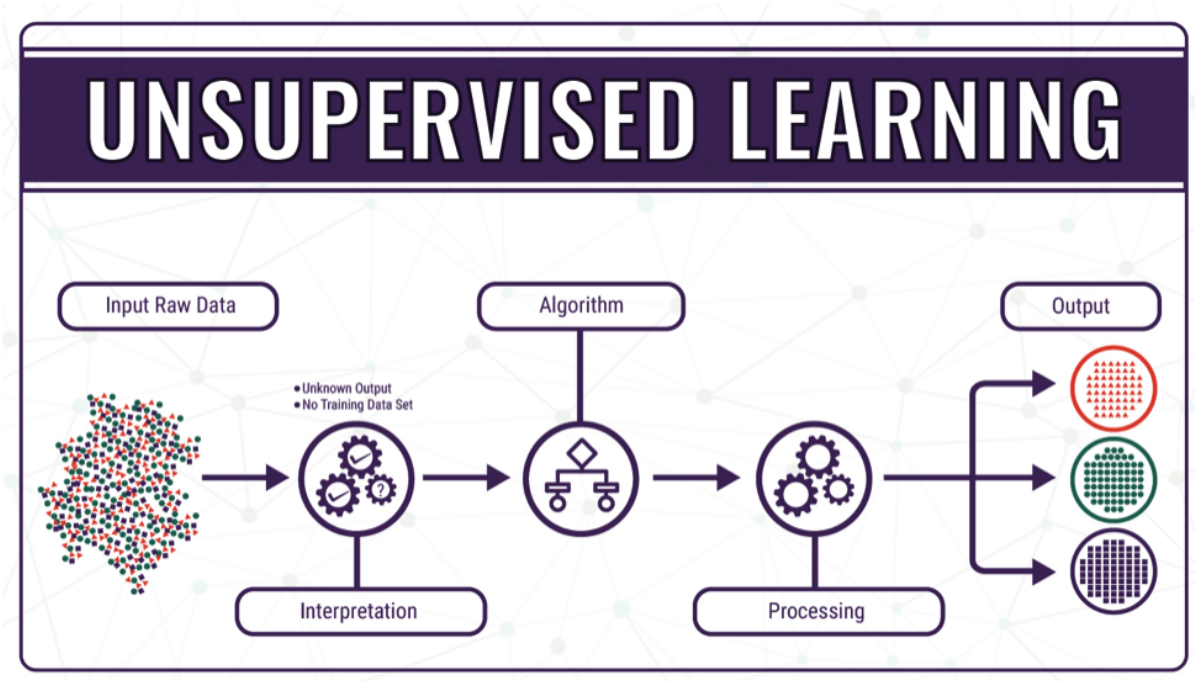
Unsupervised Learning

Definition:

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using a training dataset. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabelled data.

Unsupervised Learning Algorithms allow users to perform more complex processing tasks than supervised learning. Although, unsupervised learning can be more unpredictable than other natural learning methods.

Its ability to discover similarities and differences in information makes it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.



Why Unsupervised Learning?

Reasons for using Unsupervised Learning in Machine Learning:

- Unsupervised machine learning finds all kinds of unknown patterns in data.
- Unsupervised methods help you to find features that can be useful for categorization.
- It is taken place in real-time, so all the input data is to be analyzed and labeled in the presence of learners.
- It is easier to get unlabeled data from a computer than labeled data, which requires manual intervention.

Problems with Unsupervised Learning:

- Unsupervised Learning is more complex as compared to Supervised Learning tasks.
- It is difficult to guess if the results are correct, since there is no labeled output for testing.
- External evaluation is required.

Applications:

The following are some of the most popular real-world uses of unsupervised learning:

- **Anomaly detection**
  Unsupervised learning methods can sift through enormous volumes of data to find anomalous data points. These abnormalities might raise awareness of malfunctioning equipment, human mistakes, or security breaches.
- **Recommendation engines**
  Unsupervised learning can aid in the discovery of data trends that can be utilized to generate more successful cross-selling tactics by using historical purchase behavior data.
- **Medical Imaging**
  Unsupervised machine learning gives critical aspects to medical imaging technologies, such as image identification, classification, and segmentation, which are utilized in radiology and pathology to swiftly and effectively diagnose patients.

Categories:

Unsupervised learning models are used primarily for three kinds of problems: clustering, association, and dimensionality reduction.

- **Clustering**
  Clustering can be considered the most important unsupervised learning problem. Clustering is a technique that organizes unlabeled data into groups based on similarities and differences. Clustering techniques are used to arrange raw, unclassified data items into groups characterized by information structures or patterns.
- **Association**
  This is a rule-based approach for determining associations between variables in a given dataset. These methodologies are commonly used in market basket analysis, helping businesses better understand the linkages between various items. Recommendation engines on multiple sites are the best-known example of this.
- **Dimensionality reduction**
  While more data typically gives more accurate findings, it can also influence the performance of machine learning algorithms (overfitting) and make the visualization of datasets challenging. Dimensionality reduction is a strategy that is employed when the amount of characteristics, or dimensions, in a given dataset is excessive. It minimizes the amount of data inputs to a reasonable quantity while keeping the dataset's integrity as much as feasible.

Reinforcement Learning

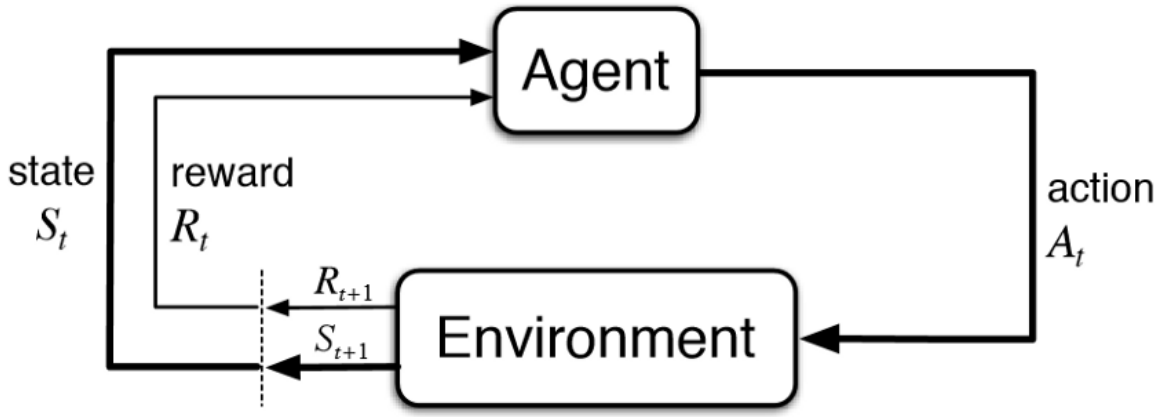
“Learn from Mistakes”

Definition:

Reinforcement Learning (RL) is the science of decision making. When compared to supervised and unsupervised learning, reinforcement learning is quite distinct. It is based on rewarding desired behaviors and punishing undesired ones.

Reinforcement Learning is defined as a Machine Learning method concerned with how software agents should take actions in an environment. Reinforcement Learning is a part of the deep learning method that helps you maximize some portion of the cumulative reward.

In Reinforcement Learning, the agent learns automatically using feedback without any labeled data, unlike supervised learning. Without a training dataset, it is bound to learn from its experience. In simple words, we can say that the output depends on the state of the current input and the following input depends on the output of the previous input.



Applications:

A possible application of RL is any real-world situation in which an agent must interact with an unpredictable environment to achieve a particular objective.

- **Autonomous driving**
  An autonomous driving system must execute many perceptual and planning tasks in an unpredictable environment. Vehicle route planning and motion prediction are two particular applications where RL might be helpful.

- **Video games**
  Learning to play video games is one of the most popular applications of reinforcement learning. Consider Google's reinforcement learning applications, AlphaZero and AlphaGo, which learned to play Go.

- **Managing resources**
  Reinforcement learning is effective in navigating complicated surroundings. It may deal with the necessity to balance various requirements. Take Google's data centers, for example. They employed reinforcement learning to balance the need to meet our power demand while being as efficient as possible, resulting in significant cost savings.

Types of Reinforcement Learning:

There are mainly two types of reinforcement learning, which are:

Positive Reinforcement:
Positive reinforcement learning means adding something to increase the expected behavior's tendency to occur again. It impacts positively on the agent's behavior and increases the strength of the behavior.
This type of reinforcement can sustain the changes for a long time, but too much positive reinforcement may lead to an overload of states that can reduce the consequences.

Negative Reinforcement:

Negative reinforcement learning is the opposite of positive reinforcement as it increases the tendency that the specific behavior will occur again by avoiding the adverse condition.
It can be more effective than positive reinforcement depending on the situation and behavior, but it provides reinforcement only to meet minimum behavior.

Supervised vs Unsupervised Learning

Supervised Learning	Unsupervised Learning
Supervised learning uses labeled input and output data.	Unsupervised learning algorithms are trained using unlabeled data.
In supervised learning, the goal is to predict outcomes for new data.	In an unsupervised learning algorithm, the goal is to get insights from large volumes of new data.
A supervised learning model produces an accurate result.	An unsupervised learning model may give less accurate results as compared to supervised learning.
Supervised learning is a simpler method.	Unsupervised learning is computationally complex.
The number of classes is known.	The number of classes is not known.
Regression and Classification are two types of supervised machine learning techniques.	Clustering and Association are two types of Unsupervised learning.
It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Clustering, KNN, and the Apriori algorithm.
Supervised learning models can be time-consuming to train, and the labels for input and output variables require expertise.	Unsupervised learning methods can have wildly inaccurate results unless you have a human intervention to validate the output variables.

Supervised learning models are ideal for spam detection, sentiment analysis, weather forecasting, and pricing predictions, among other things.	Unsupervised learning is an excellent fit for anomaly detection, recommendation engines, customer personas, and medical imaging.

Introduction to Supervised Learning

Introduction

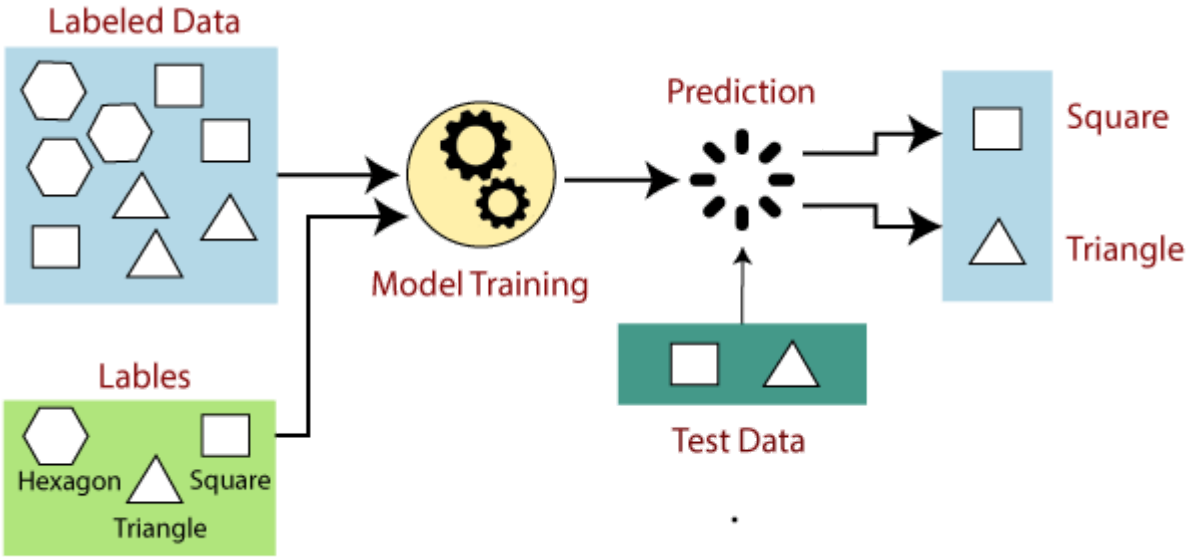
Supervised Machine Learning, more generally known as Supervised Learning, is perhaps the numerous common subbranch of Machine Learning. Professionals who begin their ML journey often begin with Supervised Learning algorithms. The phrase "supervised" learning directs to training. This algorithm is equivalent to having a teacher oversee the entire process.

Supervised Learning: How it works

The training data for a supervised learning system will include inputs coupled with the correct outputs. This data is then fed as an input to our machine. The algorithm will look for patterns in the data that correspond with the intended outcomes during training. Behind training, a supervised learning algorithm will take in unknown inputs and select which label the new inputs will be categorized based on earlier training data. A supervised learning model aims to anticipate the proper label for newly provided input data.
If that was unclear to you, let's take an example to understand the work better.
Our dataset comprises three shapes: Triangles, Squares, and Hexagons.
We'll train the model by labelling a shape as

- Triangle if it has three sides
- Square if it had four sides
- Hexagon if it has six sides

After training, we use the test set to put our model to the test, and the model's objective is to identify the shape presented to it.



Types of Supervised Learning

Supervised learning can be additionally grouped into two subproblems: classification and regression.

Classification

When the output variable is categorical, classification techniques are applied. This identifies separate entities in the dataset and tries to infer how those items should be marked or described. We'll examine this in detail in the forthcoming section.

Regression

In the domain of machine learning, regression analysis is a crucial concept. In a nutshell, the purpose of a regression model is to create a mathematical equation that describes y as a function of the x variables. Following that, using updated values for the predictor variables (x), this equation may be used to predict the outcome (y).
Let's dive into the types of regression models.

Types of Regression

Regressions of many forms are used in data science and machine learning. Each kind is essential in various contexts, but at their heart, all regression methods examine the effect of the independent variable on the dependent variables. Here are some of the most effective forms of regression:

- Linear Regression
- Logistic Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression
- Ridge Regression
- Lasso Regression

We'll be exploring some of the most important ones later.

Classification

The Classification method is a Supervised Learning technique that uses training data to identify the category of recent observations. An algorithm in Classification learns from a given dataset or observations and then classifies additional observations into one or numerous classes or categories. A classifier is an algorithm that performs classification on a dataset. Classifiers are broadly divided into two categories:

- Binary classifier: Problems that have only two possible outcomes.
- Multi-class classifier: Problems that have more than two outcomes

Types of Classification Algorithms

Classification algorithms can be divided into two main categories.

- Linear Models
    - Logistic Regression
    - Support Vector Machines
- Non-linear Models
    - K-Nearest Neighbours
    - Kernel SVM
    - Naïve Bayes
    - Decision Tree Classification
    - Random Forest Classification

Support Vector Machine(SVM)

Introduction to Support Vector Machine(SVM)

SVM is a robust supervised algorithm that works best on smaller datasets but difficult ones. Support Vector Machines, shortened as SVM, can be utilized for regression and classification tasks, but typically, they work best in classification problems. They were very prominent around the time they were completed, during the 1990s and supported as the go-to method for a high-performing algorithm with some tuning.

Table of Contents

What is a Support Vector Machine?

It is a supervised machine learning problem where we attempt to find a hyperplane that best divides the two classes. Both the algorithms try to find the best hyperplane, but the main difference is logistic regression is a probabilistic approach, whereas the support vector machine is based on statistical approaches.
Now the question is, which hyperplane does it select? There can be an infinite number of hyperplanes passing through a point and classifying the two classes perfectly. So, which one is the best?
SVM does this by finding the maximum margin between the hyperplanes, which means maximum distances between the two classes.

When to use logistic regression vs. Support vector machine?

Depending on the number of features you have, you can choose Logistic Regression or SVM.
SVM works best when the dataset is small and complex. It is usually advisable to first use logistic regression and see how it performs; if it fails to give a good accuracy, you can go for SVM without any kernel (I will talk more about kernels in the later section). Logistic regression and SVM without any kernel have similar performance, but one may be more efficient than the other, depending on your features.

Types of Support Vector Machine

Linear SVM

When the data is perfectly linearly separable, we can only use Linear SVM. Perfectly linear separable means that the data points can be classified into two classes using a single straight line(if 2D).

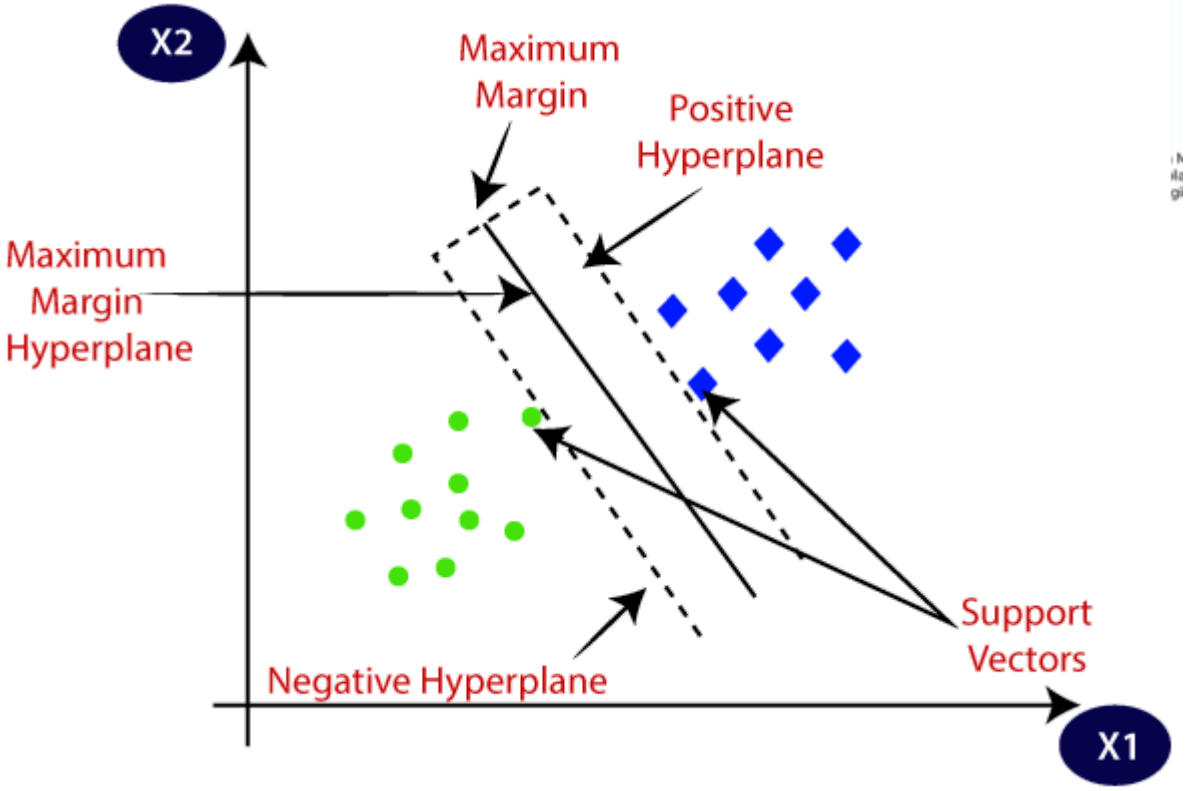
Non-Linear SVM

When the data is not linearly separable, we can use Non-Linear SVM, which means when the data points cannot be separated into two classes by using a straight line (if 2D), we use some advanced techniques like kernel tricks to classify them. We do not find linearly separable data points; hence, we use kernel tricks to solve them.

Now let's define two main terms, which will be repeated again and again in this article:

Support Vectors: These are the points that are closest to the hyperplane. A separating line will be defined with the help of these data points.

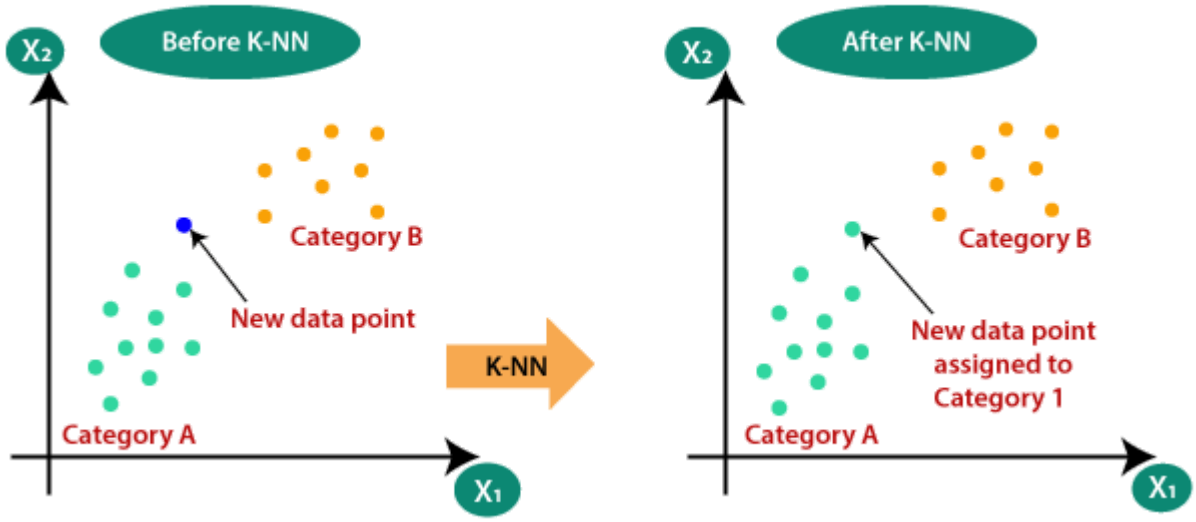
Margin is the distance between the hyperplane and the observations closest to the hyperplane (support vectors). In SVM large margin is considered a good margin.



K-Nearest Neighbor(KNN)

Why do we need a K-NN Algorithm?

Assume there are two categories, i.e., Category A and Category B, and we have a unique data point x1 so that this data point will lie in which of these categories. To solve this type of problem, we require a K-NN algorithm. With the help of K-NN, we can fast determine the category or class of a particular dataset. Consider the below diagram:



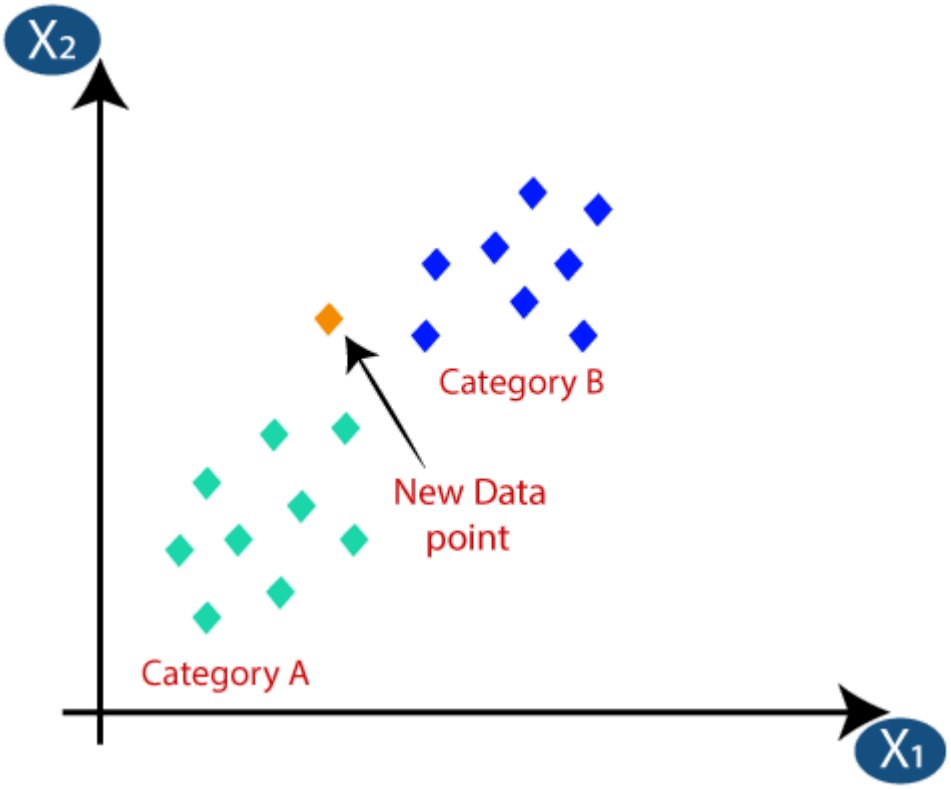
How does K-NN work?

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

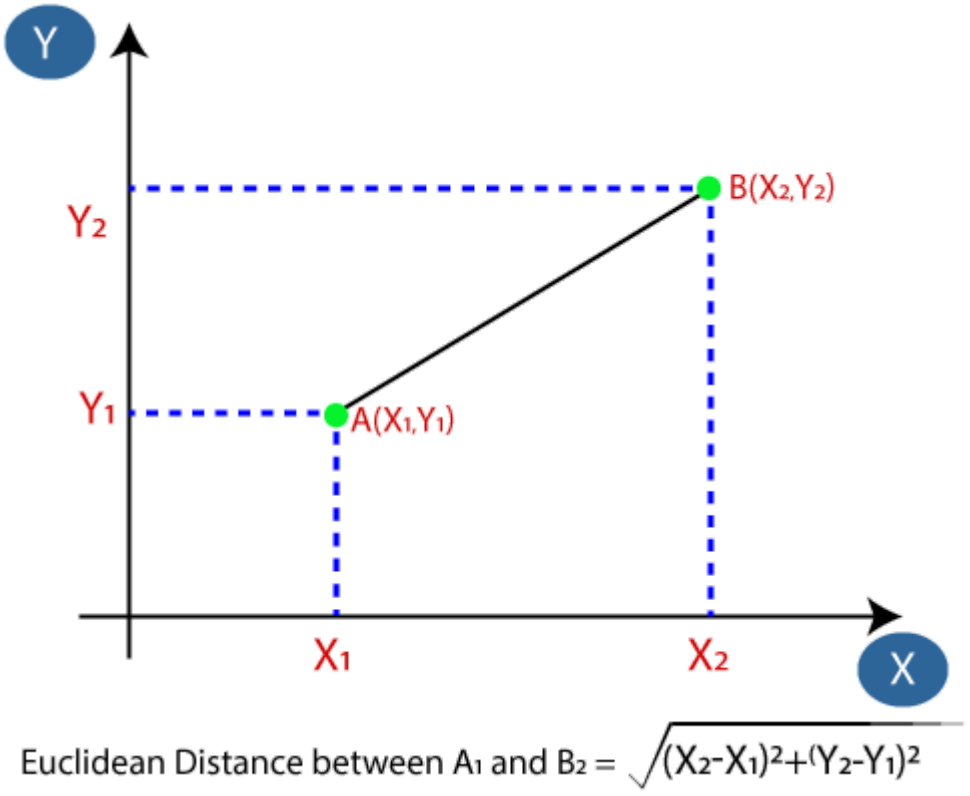
- Step-1: Select the number K of the neighbours
- Step-2: Calculate the Euclidean distance of K number of neighbours
- Step-3: Take the K nearest neighbours as per the calculated Euclidean distance.

- Step-4: Among these k neighbours, 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 neighbours is maximum.
- Step-6: Our model is ready.

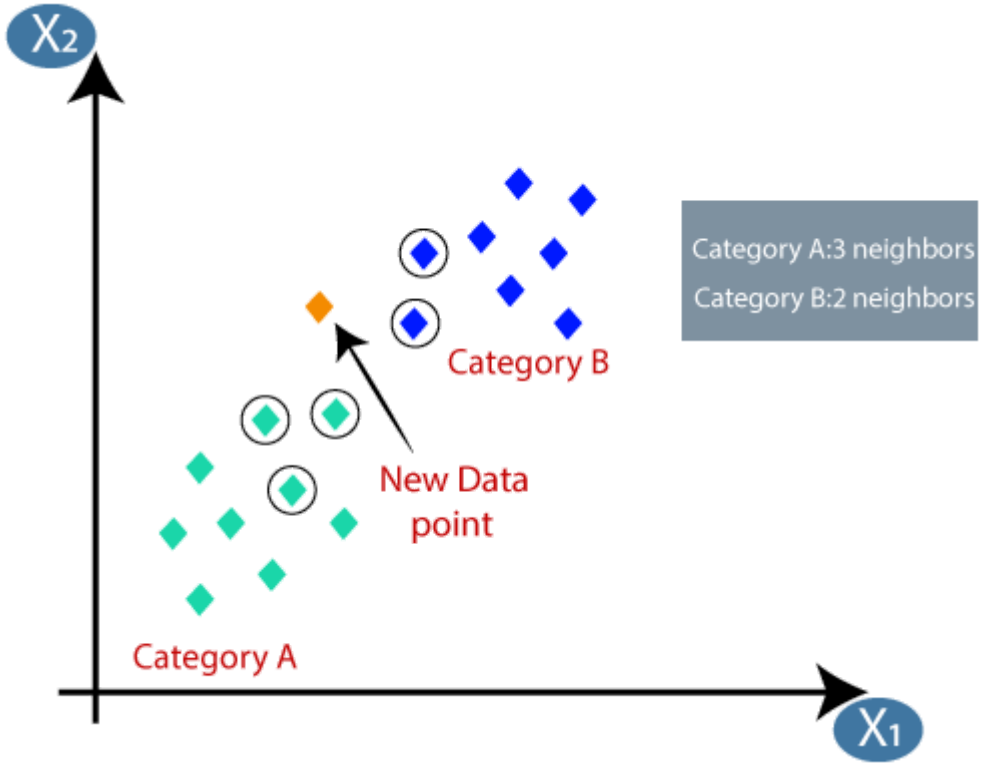
Suppose we have a new data point, and we need to put it in the required category. Consider the below image:



- Firstly, we will select the number of neighbours to choose k=5.
- Subsequent, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



- We got the nearest neighbours by calculating the Euclidean distance, three nearest neighbours in category A and two nearest neighbours in category B. Consider the below image:



- As we can see, the three nearest neighbours are from category A. Hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A meagre value for K, such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data.
- It can be more effective if the training data is significant.

Disadvantages of KNN Algorithm:

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

Interview Questions

Q1: Would you use K-NN for large datasets?

Answer :
It's not recommended to perform K-NN on large datasets, given that the computational and memory cost can increase. To understand the reason why we should remember how the K-NN algorithm works:

1. Starts by calculating the distances to all vectors in a training set and storing them.
2. Then, it sorts the calculated distances.
3. Then, we store the K nearest vectors.
4. And finally, calculate the most frequent class displayed by K nearest vectors.

So implementing K-NN on a large dataset it is not only a bad decision to store a large amount of data but it is also computationally costly to keep calculating and sorting all the values. For that reason, K-NN is not recommended and another classification algorithm like Naive Bayes or SVM is preferred in such cases.

Q2: What's the difference between k-Nearest Neighbors and Radius Nearest Neighbors?

Answer : KNN:

- The k-neighbours classification is a very commonly used technique and is widely applied in various scenarios.
- KNN implements learning based on the K nearest neighbours of each query point, where k is a hyperparameter of an integer value.
- The optimal choice of the value k is highly data-dependent: in general, a larger k suppresses the effects of noise but makes the classification boundaries less distinct.

- RNN:
  - The r-neighbours classification is used in cases where the data is not uniformly sampled or is sparse.
  - RNN implements learning based on the number of neighbours within a fixed radius r of each training point, where r is a hyperparameter of the type float.
  - The optimal fixed radius r is chosen such that points in sparser neighbourhoods use fewer nearest neighbours for the classification.

Unsupervised Learning

Introduction

Have you ever wondered how Netflix arrives to know which display will keep you connected to your laptop screens? Or you strike the gym and put on your famous Spotify track, and then Spotify automatically keeps playing channels on its own, and you still love every one of them? Well, it's not a fluke. The suggestion engines of these apps are sophisticatedly developed to provide you with products of your preference. They constantly try to learn more and more about you by keeping track of your interactions within the app and devising some underlying recurring patterns. Imagine you recently formulated a taste for the rock and roll genre. Your last four Spotify tracks have been about rock and roll. The app will probably recommend similar tracks for at least a brief period until it learns a shift in your user exercise. The technique used to design these intelligent suggestion engines is Unsupervised Machine learning.

Unsupervised Learning is one of the three broadly classified Machine Learning techniques. The objective of unsupervised Learning is to infer patterns without a target variable. Unlike supervised learning, where the objectives are well defined, i.e., we are supposed to find out the dependent or the target variable, we don't have any defined objectives in unsupervised Learning. We are supposed to extract all the relevant observations and results by critically studying the data points.

Generally known supervised learning algorithms cannot be directly applied in unsupervised learning techniques.

Some Popular Unsupervised Learning Techniques

Unsupervised learning models are employed for three major tasks:-

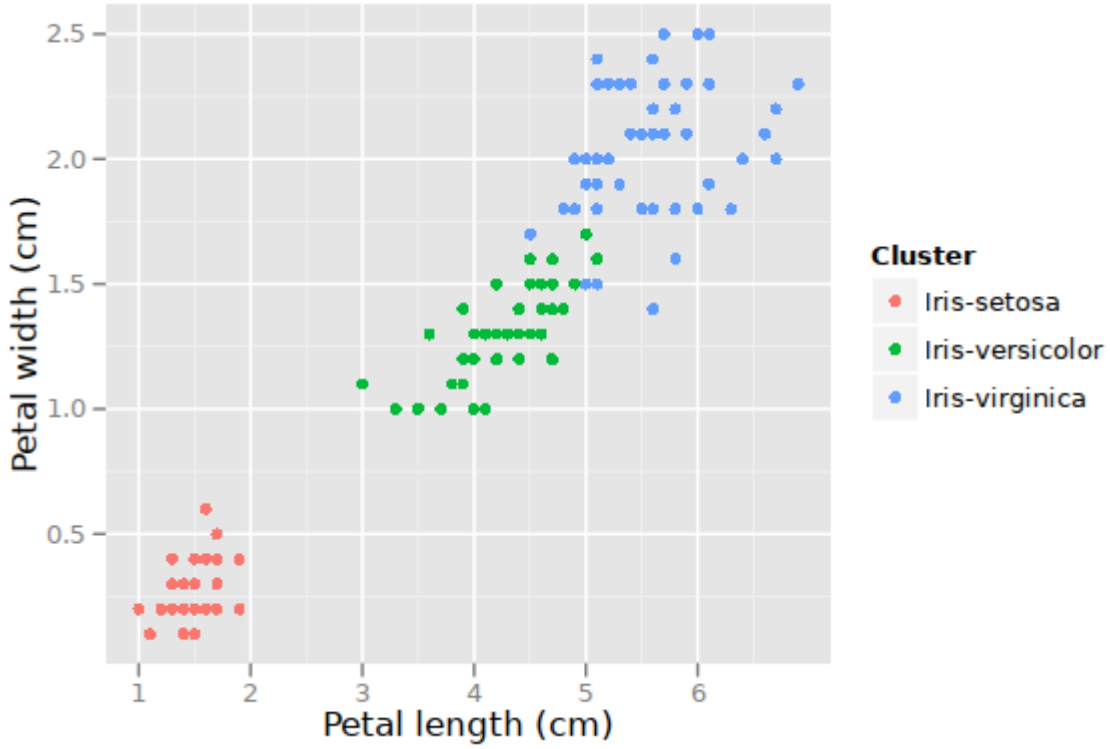
1. Clustering
2. Association
3. Dimensionality Reduction

Clustering

Grouping of equivalent data points within an unlabelled dataset is called clustering. This is one of the many famous procedures in Data Science. The data points are closely related to the other data points within their cluster. However, this strength does not mean there are no similarities between data points of different clusters. This ideal case scenario is unlikely to occur in real-world applications. But the aim should be to minimise these similarities between data points of different clusters as much as possible. Clustering is of various types:- Exclusive, Overlapping, Hierarchical, and probabilistic.

Exclusive Clustering

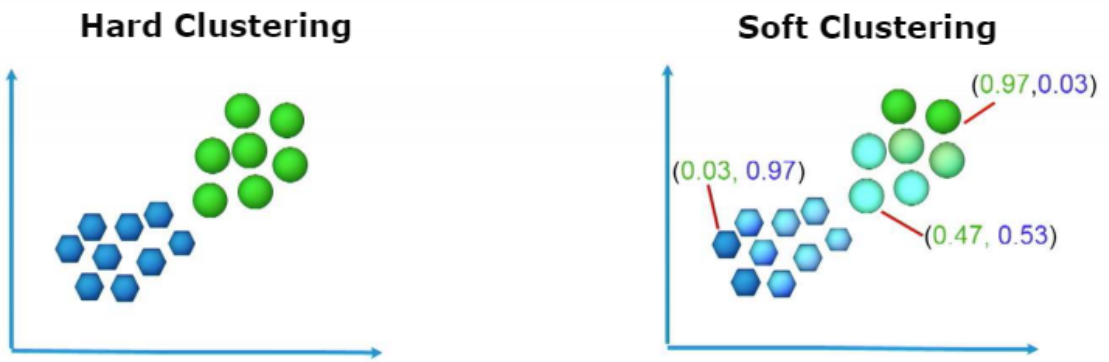
Exclusive clustering prescribes that a data point can't be shared among different clusters, i.e., any data point cannot be part of more than one cluster. This clustering technique is also referred to as "Hard clustering". K-means algorithm works on the principle of Exclusive clustering.
In K means clustering, data points are assigned into K groups (where K refers to the number of groups or clusters) depending on their distance from the centroid. More groups (higher value of K) would imply more specificity of the clusters and vice versa. It's important to remember that we need to take an optimised K value to avoid overfitting or underfitting.



Above is an implementation of K-means clustering where we have clustered flowers based on their petal length and width. The dataset used is the iris dataset which is available on the internet. We would recommend having a look at the dataset for a better understanding.

Overlapping Clustering

Sometimes It might not be possible to put a data point into one cluster. This is where Soft clustering comes in. Overlapping clustering allows data points to be shared with a degree of membership in different clusters. A well-known example of overlapping clustering is soft k means clustering.

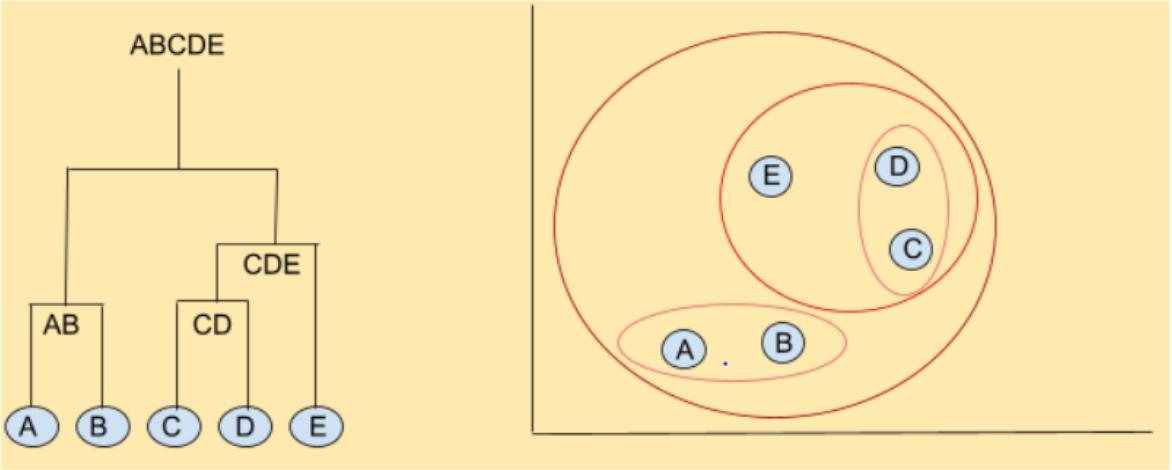


The above graphs show how Soft k-means differ from conventional or Hard k-means in clustering methods.

Hierarchical Clustering

Hierarchical Clustering can be classified in two ways:-

- **Agglomerative Clustering**:- It's a bottom-up approach to clustering the data points. Have a look at the figure below for a better understanding.



In Agglomerative Clustering, we start from individual data points, steadily building clusters (iterative clustering) hierarchically until all the data points in the data set are a part of one single cluster.

The tree-like structure depicting the hierarchy is known as a dendrogram.

These clusterings are done based on the similarity between the data points. Various methods are employed to gauge these similarities, like ward's linkage, average linkage, complete linkage, and single linkage.

- **Divisive Clustering:** Divisive Clustering is a top-down approach, the polar opposite of Agglomerative Clustering. We start from a single cluster consisting of all the data points, dividing them into smaller clusters for granularity of data patterns. Divisive Clustering isn't commonly preferred over agglomerative clustering.

Probabilistic Clustering

Probabilistic Clustering assigns data points with a probability of belonging to a specific distribution. It differs from hard clustering because it doesn't assign every data point to a particular cluster with absolute certainty. This way, it's more flexible with its clustering methods. Gaussian Method Model is the most popular technique to implement probabilistic clustering. GMMs determine in which distribution a data point belongs.

K-means Clustering

Introduction

With the rising benefit of the Internet in today's society, the quantity of data created is incomprehensibly enormous. Even though the nature of particular data is straightforward, the sheer quantity of data to be analysed makes processing complex for even computers.

To manage such procedures, we need comprehensive data analysis tools. In conjunction with machine learning, data mining methods and techniques enable us to examine enormous amounts of data intelligibly. K-means is a method for data clustering that may be utilised for unsupervised machine learning. It can classify unlabeled data into a predetermined number of clusters based on similarities (k).

Introduction to K-Means Algorithm

The K-means clustering algorithm computes centroids and reproductions until the optimal centroid is found. It is presumptively known how numerous clusters there are. It is also known as the flat clustering algorithm. The number of clusters found from data by the method is denoted by the letter 'K' in K-means.

In this method, data points are assigned to clusters so that the sum of the squared distances between the data points and the centroid is as tiny as possible. It is essential to note that reduced diversity within clusters leads to more identical data points within the same cluster.

Working of K-Means Algorithm

The following phases will allow us to comprehend how the K-Means clustering technique works-

- *Step 1:* First, we need to deliver the number of clusters, K, that need to be developed by this algorithm.
- *Step 2:* Next, choose K data points randomly and designate them separately to a cluster. Briefly, categorise the data established on the number of data points.
- *Step 3:* The cluster centroids will now be calculated.
- *Step 4:* Iterate the steps below until we find the perfect centroid, which is the appointing of data points to clusters that do not vary.
- 4.1 The sum of squared distances between data points and centroids would be estimated first.
- 4.2 At this point, we ought to allocate data points to the cluster closest to the others (centroid).
- 4.3 Finally, compute the centroids for the collections by averaging all of the cluster's data points.

K-means implements the Expectation-Maximization approach to solve the problem. The Expectation-step is used to transfer data points to the closest cluster, and the Maximization-step is used to calculate the centroid of each cluster.

When using the K-means algorithm, we must keep the following points in mind:

- It is suggested to normalise the data while trading with clustering algorithms such as K-Means since such algorithms operate distance-based measurement to determine the similarity between data points.
- Because of the iterative qualities of K-Means and the arbitrary initialisation of centroids, K-Means may become stuck in a provincial optimum and fail to converge to the global optimum. As a result, it is suggested to employ distinct centroids' initialisations.

Interview Question

Q1: What is the *Curse of Dimensionality* and how can Unsupervised Learning help with it?

Answer :

- As the amount of data required to train a model increases, it becomes harder and harder for machine learning algorithms to handle. As more features are added to the machine learning process, the more difficult the training becomes.
- In very *high-dimensional* space, supervised algorithms learn to separate points and build function approximations to make good predictions.

When the number of *features* increases, this search becomes expensive, both from a time and compute perspective. It might become impossible to find a good solution fast enough. This is the *curse of dimensionality*.

- Using *dimensionality reduction* of unsupervised learning, the most *salient* features can be discovered in the original feature set. Then the dimension of this feature set can be reduced to a more manageable number while losing very little information in the process. This will help supervised learning find the optimum function to approximate the dataset.

Q2: Give a real-life example of *Supervised Learning* and *Unsupervised Learning*

Answer :

- Supervised learning examples:
    - You get a bunch of photos with information about what is on them and then you train a model to recognize new photos.
    - You have a bunch of molecules and information about which are drugs and you train a model to answer whether a new molecule is also a drug.
    - Based on past information about spam, filtering out a new incoming email into Inbox (normal) or Junk folder (Spam)
    - Cortana or any speech automated system in your mobile phone trains your voice and then starts working based on this training.
    - Train your handwriting to the OCR system and once trained, it will be able to convert your hand-writing images into text (till some accuracy obviously)
- Unsupervised learning examples:
    - You have a bunch of photos of 6 people but without information about who is on which one and you want to divide this dataset into 6 piles, each with the photos of one individual.
    - You have molecules, part of them are drugs and part are not but you do not know which are which and you want the algorithm to discover the drugs.

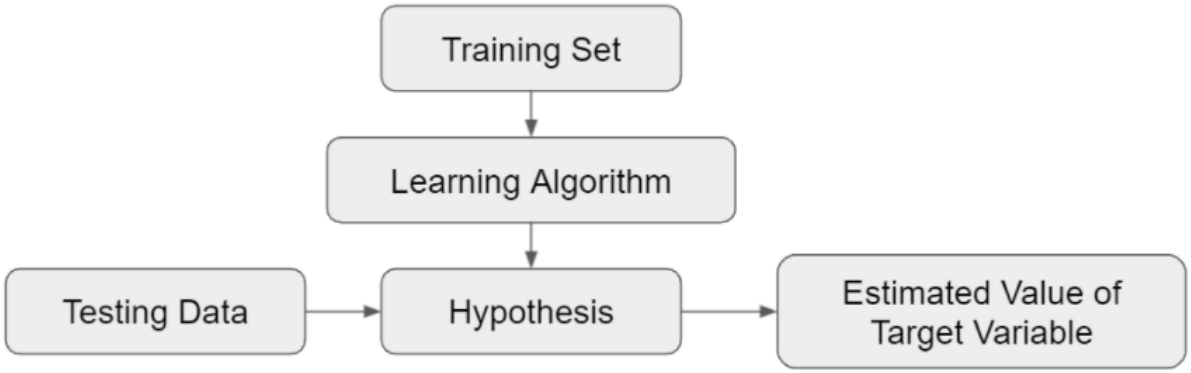
- A friend invites you to his party where you meet totally strangers. Now you will classify them using unsupervised learning (no prior knowledge) and this classification can be on the basis of gender, age group, dress, educational qualification or whatever way you would like. Why this learning is different from Supervised Learning? Since you didn't use any past/prior knowledge about people and classified them "on the go".
- NASA discovers new heavenly bodies and finds them different from previously known astronomical objects - stars, planets, asteroids, black holes etc. (i.e. it has no knowledge about these new bodies) and classifies them the way it would like to (distance from the Milky way, intensity, gravitational force, red/blue shift or whatever)

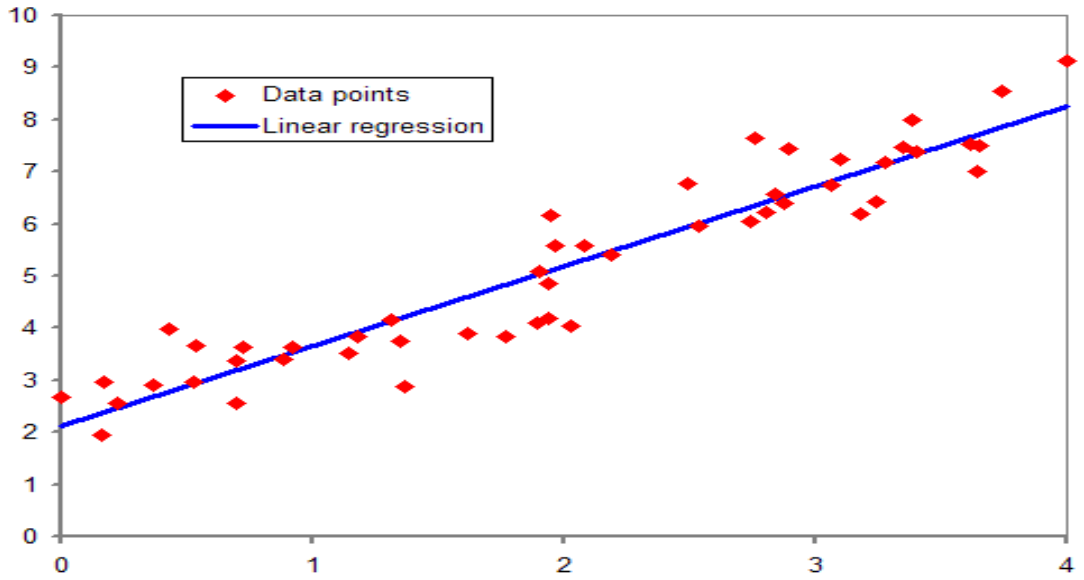
Linear Regression

What is Regression Problem

A regression problem deals with predicting actual or continuous-valued output variables such as Price. Our objective is to construct an efficient model that can foreshadow the relationship between dependent and independent variables. Out of numerous various models, linear regression is the easiest one.

Below is a Flowchart for a better acquaintance with Regression Analysis





How Linear Regression works

Divide the data into two sets, i.e., Training and Testing Data. Training *data* will train or train the ML algorithm. In distinction, *testing data* will assist validate the algorithm's anticipated output and optimizing it for better results.

There are two types of variables in Linear Regression:
Dependent variable: The outcome variable we need to anticipate is the Outcome variable.
Independent variable: We use the variable to indicate another variable's value. E.g., risk factors, predictors, or explanatory variables.

Linear Regression Analysis is forecasting the output of a variable based on the significance of other variables. The research desires to formulate a linear equation to predict the values of the dependent variable. The coefficients of the linear equation are evaluated by involving one or more independent variables. For example, Below is a simple regression problem with a single dependent and independent variable (Y and x, respectively), the form of the equation would be:
Y = C0 + C1*x (*Hypothesis*)

We obtain a successive line with a single input variable (x), but when we have multiple inputs (x), the line evolves into a Plane or hyper-plane. The complexity of linear regression rises with the number of coefficients used in the Model. The input and output variables are also known as components and target variables.

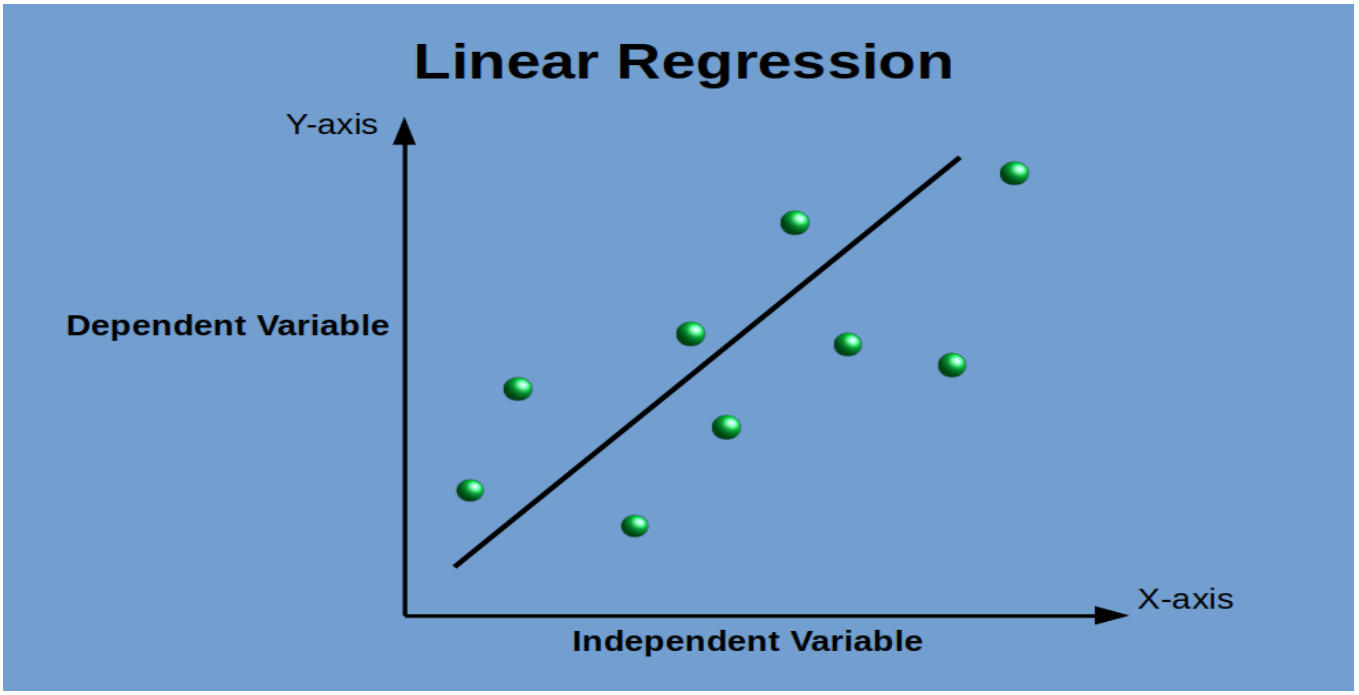
Terminologies

Before proceeding to the working linear regression algorithm, let's understand some fundamental terminologies in a straightforward linear regression.

The best fit line for easy Linear Regression will be in the form of the equation given beneath.

Y = C0 + C1*x + e

Y: Dependent variable
C0: Y-axis intercept
C1: Slope of the line
e: Error in resultant prediction



Cost Function

The regression model desires to predict Y such that the error distinction between the actual value and the anticipated value is minimum. So we update C0 and C1 to reach the best optimal value, which minimizes error to the smallest, restoring this search problem to a minimization problem.
Linear regression's cost function (J) is the Mean Squared Error between the True and Anticipated value. We square the error distinction, sum it over all the data points, and divide it by the total number of data points.

$$minimize \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

Logistic Regression

Introduction

Have you ever wondered how your email account accurately segregates regular, essential, and spam emails? It's not a complicated trick, and we'll learn its secret. This is done with a supervised learning model called Logistic regression.

Logistic regression is operated in supervised learning tasks. Additionally mainly, it is used for classification tasks. We understand that name throws some people off. But the regression in the logistic regression is just misleading. It is NOT a regression model. Logistic regression is a probabilistic function. That means it makes use of probabilities of events to make its prediction.

Methodology

Suppose we are given a task, say we are given a customer's banking history and are tasked to find if the customer can be sanctioned a loan. If given a loan, we need to find whether the customer will default on payment or not. We can use logistic regression for this purpose. It will be a binary classification between 'Yes' and 'No'. Logistic regression makes use of a sigmoid function, and it is of the form -

We know the straight line equation -

$y = w_0 + w_1x$

We know the sigmoid function has a range between 0 and 1. So let's divide the above equation by 1-y.

$y / (1-y)$: 0 for $y = 0$ and ∞ for $y = 1$

But we require our function to be between $-\infty$ to $+\infty$. For that, we'll take the logarithm so the new equation is:

$\text{Log } (y / (1-y)) = w_0 + w_1x$

Upon simplifying, our final equation then becomes -

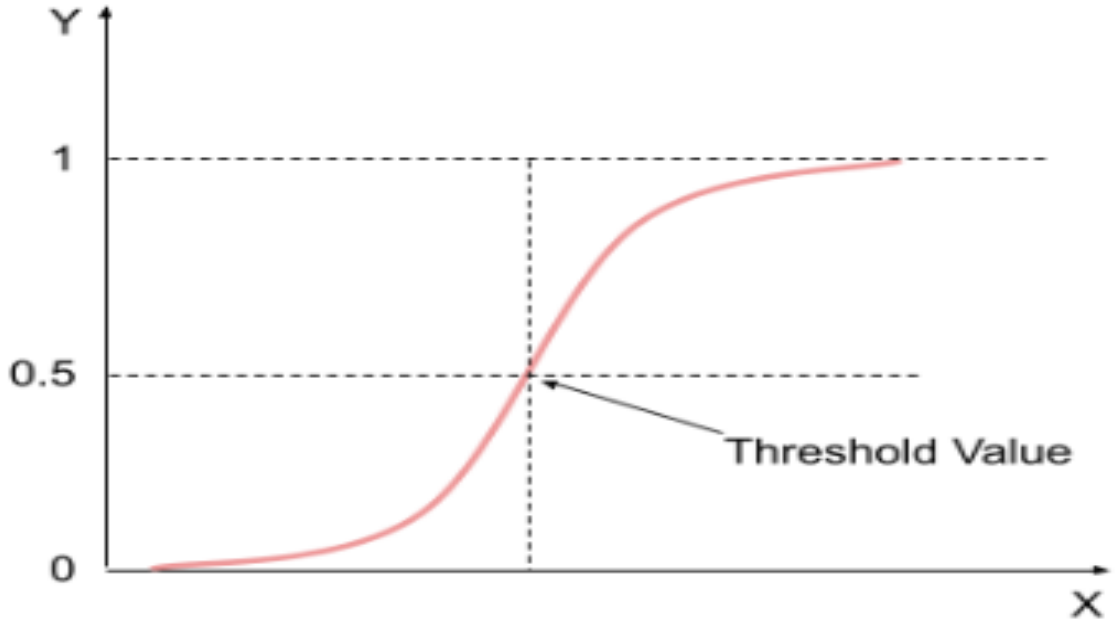
$$y = \frac{1}{1+e^{-(w_o + w_1 x)}}$$

Here y = predicted probability belonging to the default class(default class is 1(yes))

$w_0 + w_1x$ = the linear model within logistic regression.

Also, the function is in the form of a sigmoid function

The Sigmoid function has a range between 0 and 1. And therefore forms an S-like curve.



The logistic function predicts the probability of an outcome. Hence its value lies anywhere between 0 and 1. And that's where it gets its name from. We choose a threshold value above which the final prediction would be 1 and 0 otherwise.

Let's talk about the linear equation $w_0 + w_1x$ within the logistic function. Why do we need the logistic regression function in the first place if it stems from linear regression?

It's because the linear regression equation isn't confined within a range, unlike logistic regression. And it would be a very difficult task to assign a threshold value for class membership for a linear regression function. Thus we feed the predicted value to a sigmoid function which makes it Logistic regression having a range between 0 and 1. Now since the range is between 0 and 1(no outliers) it would be convenient to do a probabilistic classification.

It represents a linear relationship between the input features and the final output.

Here x = input feature

w_0 = bias term

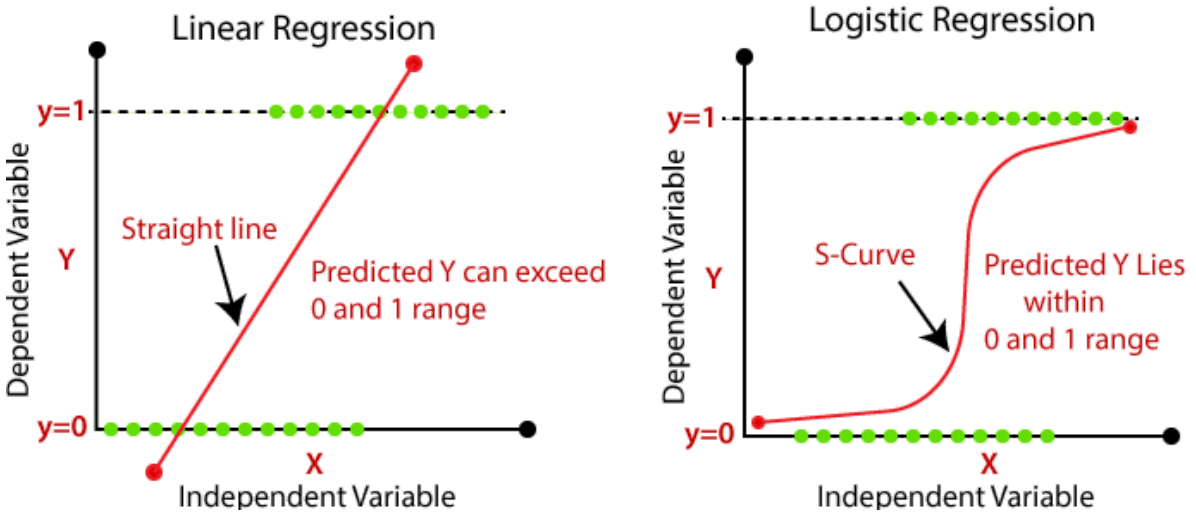
w_1 = weight associated with the input variable

Now suppose we take 0.5 as our threshold value. That means

A predicted value >0.5 from the logistic function would have the final prediction as 1 and,

A predicted value ≤0.5 from the logistic function would have the final prediction as 0.

This is also called the decision boundary.



Interview Questions

1. What is a Linear Regression?

In simple terms, linear regression is adopting a linear approach to modeling the relationship between a dependent variable (scalar response) and one or more independent variables (explanatory variables). In case you have one explanatory variable, you call it a simple linear regression. In case you have more than one independent variable, you refer to the process as multiple linear regressions.

2. Can you list out the critical assumptions of linear regression?

There are three crucial assumptions one has to make in linear regression. They are,

- It is imperative to have a linear relationship between the dependent and independent A scatter plot can prove handy to check out this fact.
- The independent variables in the dataset should not exhibit any multi-collinearity. In case they do, it should be at the barest minimum. There should be a restriction on their value depending on the domain requirement.
- Homoscedasticity is one of the most critical It states that there should be an equal distribution of errors.

3. What is Heteroscedasticity?

Heteroscedasticity is the exact opposite of homoscedasticity. It entails that there is no equal distribution of the error terms. You use a log function to rectify this phenomenon.

4. What is the primary difference between R square and adjusted R square?

In linear regression, you use both these values for model validation. However, there is a clear distinction between the two. R square accounts for the variation of all independent variables on the dependent variable. In other words, it considers each independent variable for explaining the variation. In the case of Adjusted R square, it accounts for the significant variables alone for indicating the percentage of variation in the model. By significant, we refer to the P values less than 0.05.

5. Can you list out the formulas to find RMSE and MSE?

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

What is Classification Algorithm

The **classes** are often referred to as target, label, or categories.

Classification is the process of predicting the class of given data points. **Classification** uses an algorithm to assign test data into specific categories accurately. It recognizes particular entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined. A classification problem is when the output variable is a category, such as “Red” or “blue” or “disease” and “no disease.”

A typical job of machine learning algorithms is to recognize objects and to be able to separate them into categories. This process is called classification, and it helps us segregate vast quantities of data into discrete values, i.e., distinct, like 0/1, True/False, or a pre-defined output label class.

Blood Cancer disease detection can be identified as a classification problem. This is a binary classification since there can be only two classes, i.e., has blood cancer or does not have blood cancer. The classifier, in this case, needs training data to understand how the given input variables are related to the class. And once the classifier is trained accurately, it can be used to detect whether blood cancer is there or not for a particular patient.

Some of the Classification Algorithms are:-

- Logistic Regression
- Decision Tree
- Random Forest
- SVM
- Naive Bayes

Logistic Regression

Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set.
Logistic regression is used in supervised learning tasks. More specifically, it is used for classification tasks. We know that name throws some people off. But the regression in the logistic regression is slightly misleading. It is NOT a regression model. Logistic regression is a probabilistic function. That means it makes use of probabilities of events to make its prediction.

Logistic Regression is used when the dependent variable(target) is categorical. For example,

- To predict whether an email is a spam (1) or (0)
- Whether the cancer is malignant (1) or not (0)

A logistic approach fits best when the machine's task is learning is based on two values or a binary classification. And, as more data is provided, it could know how to do this better over time.

Methodology:

Suppose we are given a task, say we are given a customer's banking history and are tasked to find if the customer can be sanctioned a loan. Basically we need to find if given a loan, will the customer default on payment or not. We can use logistic regression for this purpose. It will be a binary classification between 'Yes' and 'No.'

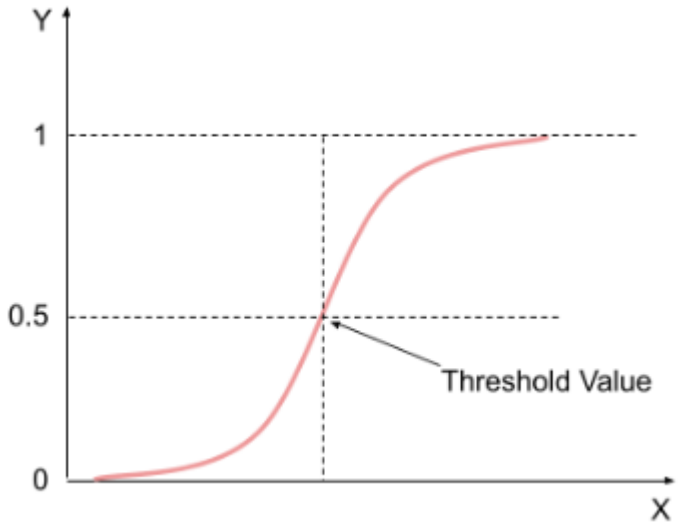
Logistic regression makes use of a sigmoid function, and it is of the form -

$$y = \frac{1}{1+e^{-(w_o + w_1 x)}}$$

Here y = predicted probability belonging to the default class(default class is 1(yes))

$w_0 + w_1x$ = the linear model within logistic regression.

The Sigmoid function has a range between 0 and 1. And therefore, it forms an S-like curve.



The logistic function predicts the probability of an outcome. Hence its value lies anywhere between 0 and 1. And that's where it gets its name from. We choose a threshold value above which the final prediction would be 1 and 0 otherwise.

Let's talk about the linear equation $w_0 + w_1x$ within the logistic function. Why do we need the logistic regression function in the first place if it stems from linear regression?

The linear regression equation isn't confined within a range, unlike logistic regression. And it would be a complicated task to assign a threshold value for class membership for a linear regression function. Thus we feed the predicted value to a sigmoid function which makes it Logistic regression between 0 and 1. Since the range is between 0 and 1(no outliers), it would be convenient to do a probabilistic classification.

It represents a linear relationship between the input features and the final output.

Here x = input feature

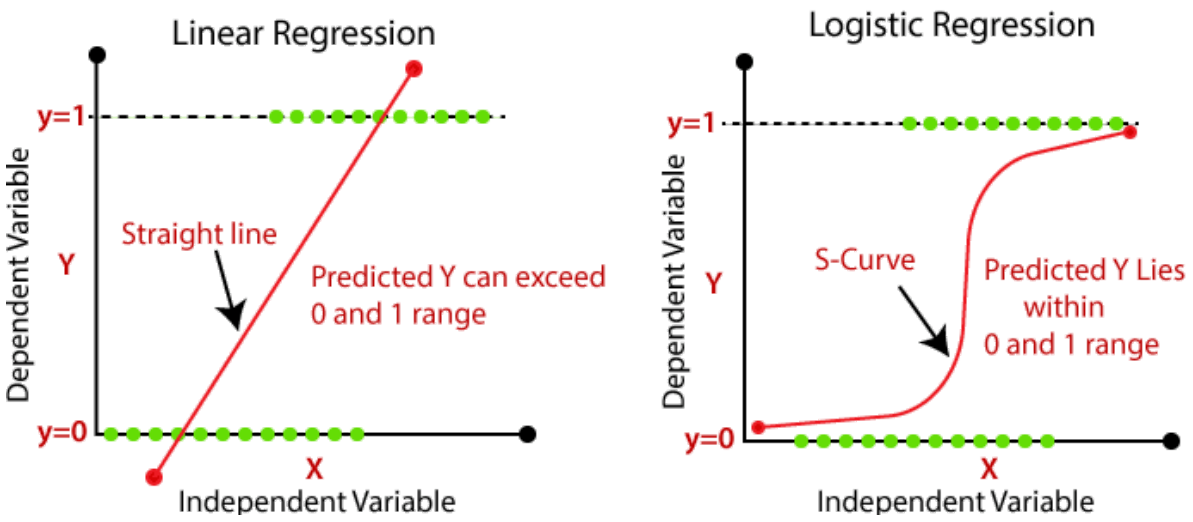
w_0 = bias term

w_1 = weight associated with the input variable

Now suppose we take 0.5 as our threshold value. That means,

A predicted value > 0.5 from the logistic function would have the final prediction as 1 and,
A predicted value ≤ 0.5 from the logistic function would have the final prediction as 0.

This is also called the decision boundary.



Plotting the graph clears what makes logistic regression different from linear regression.

Decision Tree

Decision Trees are the Supervised Machine learning algorithm that can be used for Classification and Regression problems. A decision tree is a step-by-step decision that eventually takes us to a leaf.

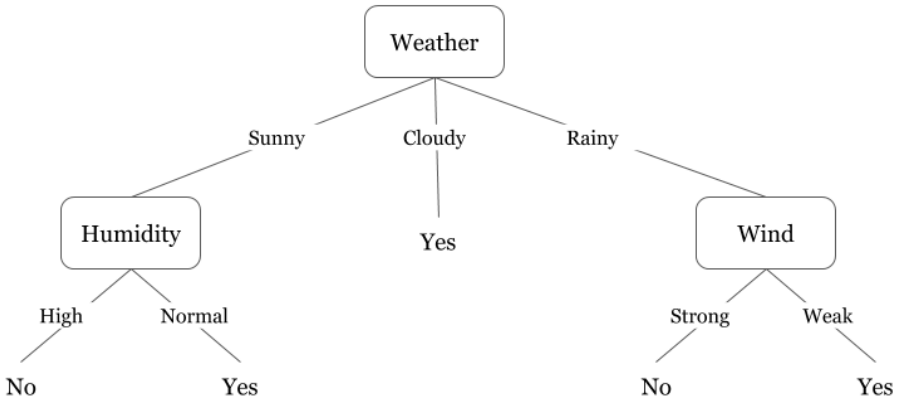
At every leaf, we have an output, and at every non-leaf node are asking the question and making decisions. The decisions or the test are performed based on features of the given dataset.

Let's illustrate this with the help of an example:- Let's assume we want to play badminton on a particular day — say Saturday — how will you decide whether to play or not. Let's say you go out and check if it's hot or cold, check the speed of the wind and humidity, and how the weather is, i.e., sunny, cloudy, or rainy. You take all these factors into account to decide if you want to play or not.
So, you calculate all these factors for the last ten days and form a lookup table like the one below.

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes

10	Rainy	Mild	High	Strong	No

Now, you may use this table to decide whether to play or not. But what if the weather pattern on Saturday does not match any of the rows in the table? This may be a problem. A decision tree would be a great way to represent data like this because it considers all the possible paths that can lead to the final decision by following a tree-like structure.



Types of decision trees are based on the type of target variable we have. It can be of two types:

1. **Categorical Variable Decision Tree:** A decision tree has a categorical target variable called a Categorical variable decision tree.
2. **Continuous Variable Decision Tree:** A decision tree has a continuous target variable. Then it is called a Continuous Variable Decision Tree.

Assumptions while creating Decision Tree:

Below are some of the assumptions we make while using a Decision tree:

- In the beginning, the whole training set is considered as the root.
- Feature values are preferred to be categorical. If the values are continuous, they are discretized prior to building the model.
- Records are distributed recursively based on attribute values.
- Order to place attributes as root or internal node of the tree is done using some statistical approach.

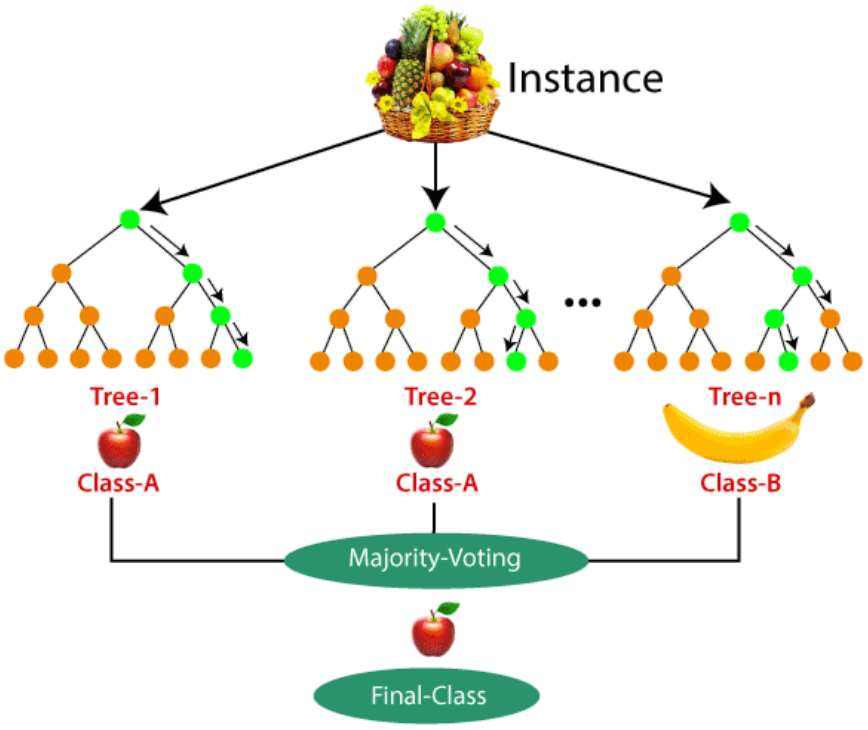
Random Forest

Random Forest is a supervised Machine Learning Algorithm, the extended version of the Decision Tree.

The problem that arises in the Decision Tree is Overfitting. The algorithm tries to fit the sample data according to the sample output accurately and performs well on the sample data. But may perform very poorly in the case of the testing data.

Random Forest says that let's not relied on only one Decision Tree. Instead, we will use multiple Decision Trees and select the majority result out of them. We will use the same training examples to build numerous Decision Trees, but they all will be different from each other.

So, how is this possible?
We will not use the same features to build the Decision Tree but will use some randomness to get to the point where one outlier might affect a few of the Decision Trees, but not all of the Decision Trees.



Methodology:

In Random Forest, we use multiple classifiers. The data fed in each classifier should be different from each other. We use a method called **Bagging**.

Bagging is the shortcut of the Bootstrap Aggregation Algorithm. Bagging says that if we have 'n' data points, let's select 'm' out of it, but with replacement, i.e., one data point can come multiple times. In this way, if any data point is present more than once, some of them must be missing.

Along with the Bagging, we make Feature Selection as well. In Feature selection, we will not train a Decision Tree based on all the features. If there are 'n' features, let's select 'k's out of it randomly. But with no replacement. The very standard number to choose the quality is n out of n features. In this way, each Decision Tree is trained on different features and different training datasets. The good part about the feature selection is that if there is one feature that is creating a problem for us, we will have some trees that will not have that particular feature and hopefully will be able to reduce the overfitting problem.

Advantages of Random Forest:

- Preprocessing is not required. Missing values are handled.
- The Random Forest algorithm is Less prone to overfitting than the Decision Tree and other algorithms.
- The Random Forest algorithm Outputs the very useful importance of features.
- The Random Forest algorithm can be used as a dimensionality reduction technique.
- The Random Forest algorithm can handle large datasets with higher dimensionality.

Disadvantages of Random Forest:

- The Random Forest algorithms may change considerably by small changes in the data.
- The Random Forest algorithm calculation can go more complex than the other algorithms.
- The Random Forest algorithm requires more time to train the model as many trees are involved.
- The Random Forest algorithm is not very suitable for regression analysis.
- The Random Forest algorithm is relatively less interpretable than the Decision Tree and other algorithms.

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy. Even for a large dataset, it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

Naive Bayes

Naive Bayes is a set of simple and efficient machine learning algorithms for solving various classification and regression problems.

What is Bayes Theorem?

Bayes Theorem is a mathematical representation for determining conditional probability. It is the probability of an event occurring based on the previous occurring outcome. Bayes theorem is also called Bayes Rule or Bayes Law and is the foundation of Bayesian statistics. The mathematical formula for Bayes Theorem is:

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

P(A/B) - the probability of event A occurring, given event B has already occurred.
P(B/A) - the probability of event B occurring, given event A has already occurred.
P(A) - the probability of event A occurring
P(B) - the possibility of event B occurring

How does the Naive Bayes algorithm work?

Let us understand the functioning of the Naive Bayes algorithm using an example. Below is a training dataset of weather and corresponding target

variable Play. Now we need to classify whether we should play or not based on the weather conditions.

Whether	Play
Sunny	No
Sunny	No
Overcast	Yes
Rainy	Yes
Rainy	Yes
Rainy	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table

Whether	No	Yes
Overcast		4
Sunny	2	3
Rainy	3	2
Total	5	9

Likelihood Table 1

Whether	No	Yes		
Overcast		4	=4/14	0.29
Sunny	2	3	=5/14	0.36
Rainy	3	2	=5/14	0.36
Total	5	9		
	=5/14	=9/14		
	0.36	0.64		

Likelihood Table 2

Whether	No	Yes	Posterior Probability for No	Posterior Probability for Yes
Overcast		4	0/5=0	4/9=0.44
Sunny	2	3	2/5=0.4	3/9=0.33
Rainy	3	2	3/5=0.6	2/9=0.22
Total	5	9		

The first step is to construct a frequency table of the given data. Next, we will create a likelihood table by finding the probabilities of sunny, rainy, etc. The final step is to use the Naive Bayes equation and find the probability of each category.

Problem statement: Players will play if the weather is sunny. Is this the correct statement?

We'll solve the question by using the Naive Bayes formula.

P(Yes / Sunny) = P(Sunny / Yes) * P(Yes) / P (Sunny)
Here we have,
P (Sunny /Yes) = 3/9 = 0.33,
P(Sunny) = 5/14 = 0.36,
P(Yes)= 9/14 = 0.64
Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60
which has a higher probability.

There are three categories of Naive Bayes models.

1. Gaussian Naive Bayes
2. Multinomial Naive Bayes
3. Bernouli Naive Bayes

Advantages of Naive Bayes Classifier

- Naive Bayes is one of the most fast-moving and effortless Machine Learning algorithms to predict the class of a dataset.
- It can manage to perform Binary as well as Multi-class Classifications.
- It is best suited for multi-class predictions as compared to any other algorithm.
- The most popular practical implementation of this classifier is for text classification problems.

Disadvantages of Naive Bayes Classifier

There are various disadvantages of Naive Bayesian classification,

- The data may not always be independent of each other.
- This algorithm can not be used for an imbalanced dataset.
- When we encounter a response vector in the test data for a particular class absent in the training data, we might end up with zero class probabilities; this is known as the Zero probability problem.

Applications of Naive Bayes Classifier

Naive Bayes Classifier has multiple real-life applications, to name a few are:

- Spam filtration
- Text classification
- Sentiment analysis
- Recommendation System
- Multi-class prediction

Interview Questions

1. **Why is logistic regression called regression and not classification?**
   The main difference between regression and classification is that the output variable in the regression is numerical (or continuous) while that for classification is categorical (or discrete). Logistic regression is basically a supervised classification algorithm. However, the model builds a regression model just like linear regression to predict the probability that a given data entry belongs to the category numbered as “1”.

2. **Which kind of problems are decision trees most suitable for?**
   - Decision trees are most suitable for tabular data.
   - The outputs are discrete.
   - Explanations for decisions are required.
   - The training data may contain errors.
   - The training data may contain missing attribute values.

3. **Can Random Forest Algorithm be used both for Continuous and Categorical Target Variables?**
   Yes, Random Forest can be used for both continuous and categorical target (dependent) variables.
   In a random forest, i.e., the combination of decision trees, the classification model refers to the categorical dependent variable, and the regression model refers to the numeric or continuous dependent variable.

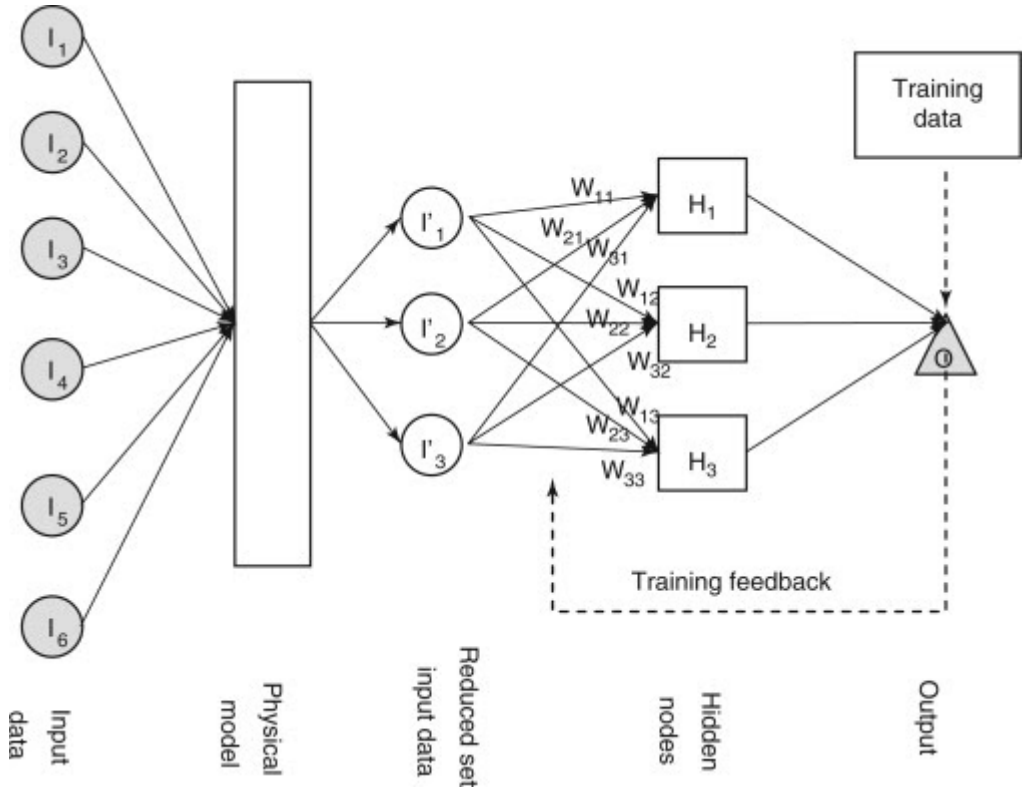
4. **How does Naive Bayes work?**
   It calculates two probabilities: the probability for each class and the conditional probability for each class according to some condition. All these probabilities are calculated for the training data, and after training, new data points can be predicted using the Bayes theorem. Naive Bias can also be trained in a semi-supervised manner using a mixture of the labeled and unlabelled datasets.

5. **Why is naive Bayes so ‘naive’?**
   Naive Bayes is so ‘naive’ because it assumes that all of the features in a data set are equally important and independent. As we know, these assumptions are rarely true in real-world scenarios.

Artificial Neural Network

Introduction to Artificial Neural Network

- The best way to understand ANN and its working at the backend is to understand how you learn and perceive the surroundings around you.
- Remember your first time riding a bicycle? A countless number of iterations of riding, falling, and getting back up again, learning some minute details from your errors for the next time you hit the pedal. At a higher level, it is what ANN is all about. Executing a task and learning from mistakes.
- ANNs simulate the neural network in humans. A human neural network consists of billions of neurons. It's these neurons that give humans the ability of learning and recognise things from their past experiences. These neurons work closely with each other towards a common objective. ANN simulate this neural structure of a human brain with a layered structure, which basically consists of three major components- The input layer, the hidden layer, and The output layer.



The input layer

The input layer is the first component in the architecture of an ANN. It receives the input for various explanatory attributes and includes the bias term as well. So say we have n input variables or attributes, the size of the input layer will be n+1 where the extra variable is the bias term.

The hidden layer

This is where the majority of computation happens. It receives the processed data from the input layer. This data is nothing but the input variables. Now, It may or may not be true that all these variables are equally essential in computing the final predictions. For example, let's say we want to train an ANN model to recognise Siberian tigers. So there are some defining features about a Siberian tiger that would be very crucial for the predictions. Like their golden fur with prominent dark stripes and the canines. These features would be given more importance over other

features of the input. Now there might be relatively less crucial features like the ears or the tail. They may be used to make the prediction but wouldn't be considered defining features of a tiger. So to differentiate between the variables based on the impact they may have on the final predictions, we assign weights to these variables. Initially, random weights are assigned but these weights as we go along. This process of adjusting the weights is called 'Backward propagation for errors' or more commonly known as 'Backpropagation'. We'll discuss this in detail later in the blog.

The output layer

The final component in the ANN architecture. Finally, the hidden layers link to the 'output layer'. The output layer receives connections from hidden layers or from the input layer. It returns output corresponding to the input variables. The active nodes in the output layer combine and change values in the data to produce output values. The key to an effective ANN lies in the appropriate selection of weights.

Some major processes in ANN

Forward Propagation

As the data moves from the input layer to the hidden layer and then to the output layer, computing the output for each iteration in the training phase, the process is called Forward Propagation.

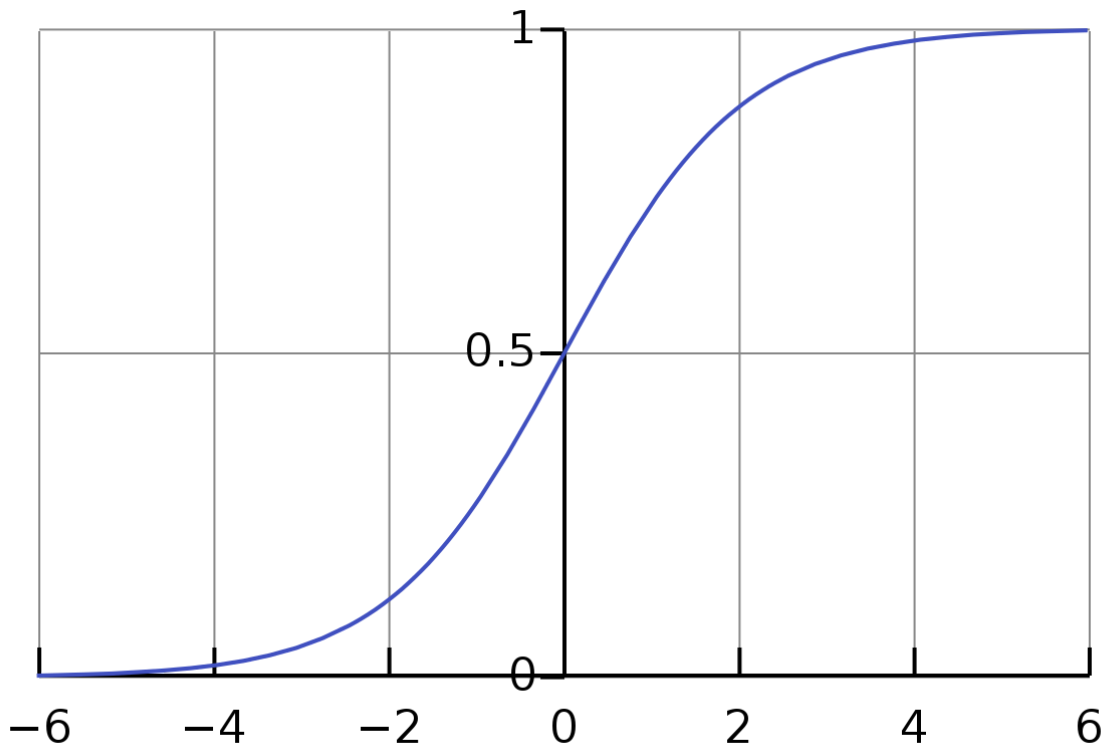
Backpropagation

We learned that we assign weights to features based on the impact they have on the final prediction. Initially, we assign random values to the weights and after each iteration(or forward propagation), the cost function is used to revise these weights. In a way, it works as a feedback system where we make our prediction which is then

validated by the actual output to see how close or far the model is from the actual prediction.

The activation function

Activation functions are mathematical equations that convert the output of each layer before it is passed to the next consecutive layer. It alters the output from the layers in a way that makes it more convenient for predictions. The most commonly used activation function is the sigmoid function which has a value between 0 and 1.



Advantages and Disadvantages of ANN

1. ANN models are extremely versatile. They can be used to perform Linear and non-linear tasks alike. But this comes with its own challenges. One major challenge to ANN is the high volume of data that needs to be fed for training. It's because an efficient model requires a variety of data to

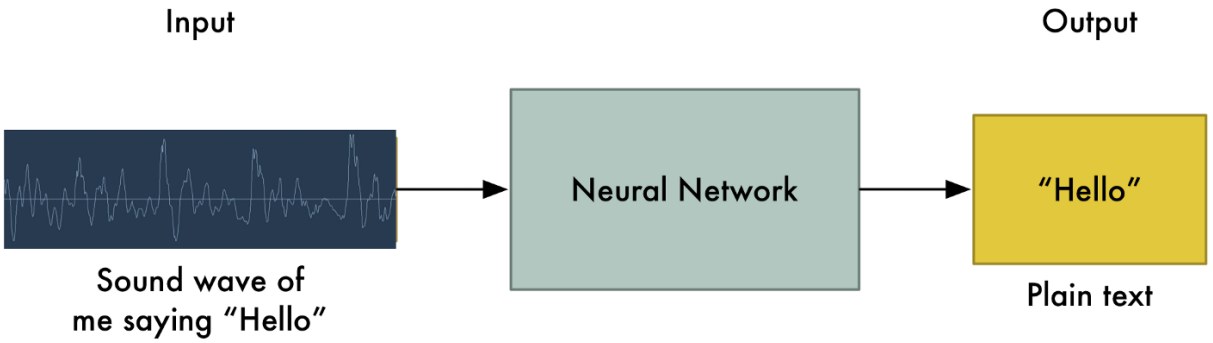
encompass all the varying input types and effectively learn for the task a model is aimed for.

2. ANNs aren't entirely dependent on each and every unit. If a few of them fail to respond, an ANN may still be able to produce the required output. But this requires a huge computation cost, in terms of storage as well as processing power.

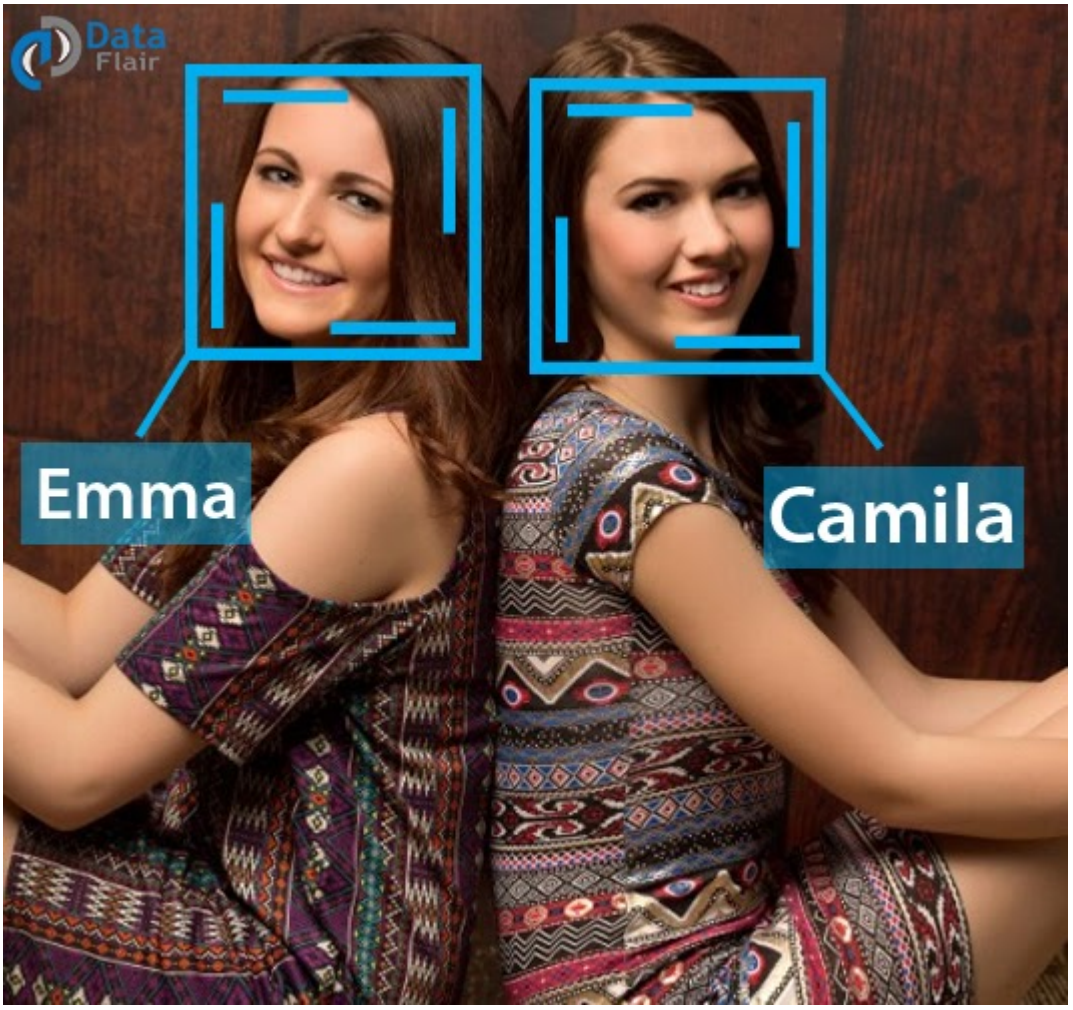
3. Every unit in ANNs need not be explicitly programmed for the task it is supposed to take. It's very autonomous in task allocation and learning objectives. However, this may lead to the user being absolutely clueless about how exactly the model is making the predictions it makes. The implementation among the nodes in the network may be vague to the user himself.

Applications of ANN

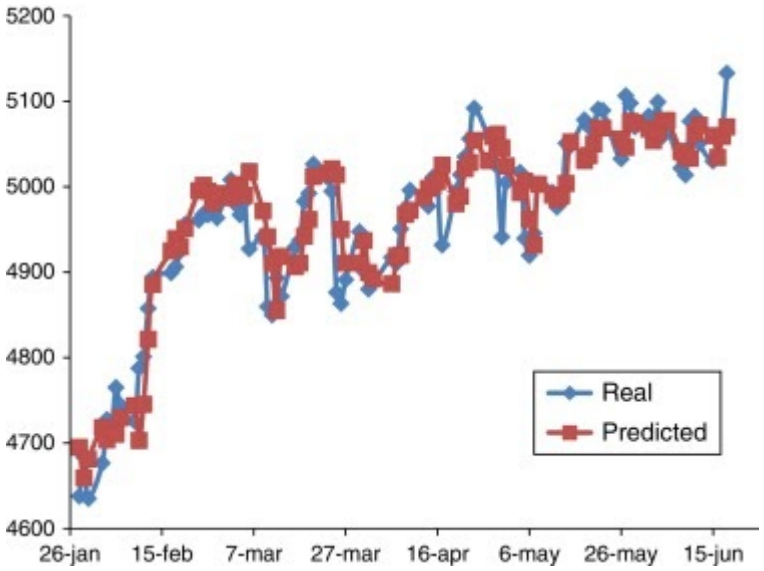
1. **Speech recognition:-** With deep learning, Speech recognition can be effectively implemented. Earlier we had other methods which were based on statistical data like the Hidden Markov model but they weren't as efficient as implemented by ANN.



2. **Facial recognition:-** One of the most commonly known examples of Deep learning. Facial recognition systems can be efficiently implemented with Convolutional neural networks. Facial recognition is a very common feature in smartphones these days.



3. **Stock market prediction:-** A stock market predictor can be efficiently implemented with a multilayer perceptron. MLPs consist of several layers of nodes each of which is fully connected with each node in the next layer. The input variables may be the opening price, past performance, annual returns, etc.



Convolutional Neural Network

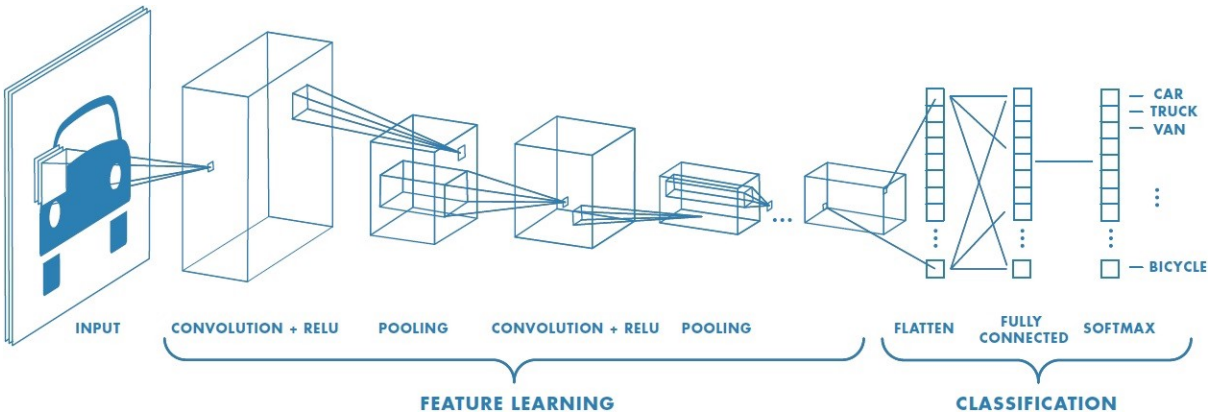
Introduction to CNN

- The convolutional Neural Network CNN operates by obtaining an image, designating it some weightage established on the various objects of the image, and then differentiating them from each other.
- CNN needs very little pre-process data as corresponded to other deep learning algorithms. One of the primary abilities of CNN is that it applies elementary methods for training its classifiers, which makes it reasonable sufficiently to learn the characteristics of the target object.
- CNN is based on analogous architecture, as found in the human brain's neurons, specifically the Visual Cortex. Each neuron responds to a particular stimulus in a specific region of the visual area identified as the Receptive field. These collections overlap to contain the whole visual area.

What is CNN?

- A CNN is one of the most influential Deep Learning algorithms that can handle an input image, allocate importance to various aspects of the picture, and distinguish one from another. The preprocessing needed in a CNN is much lower than in various classification algorithms.
- The architecture of CNN is analogous to that of the connectivity pattern of Neurons in the Human Brain.

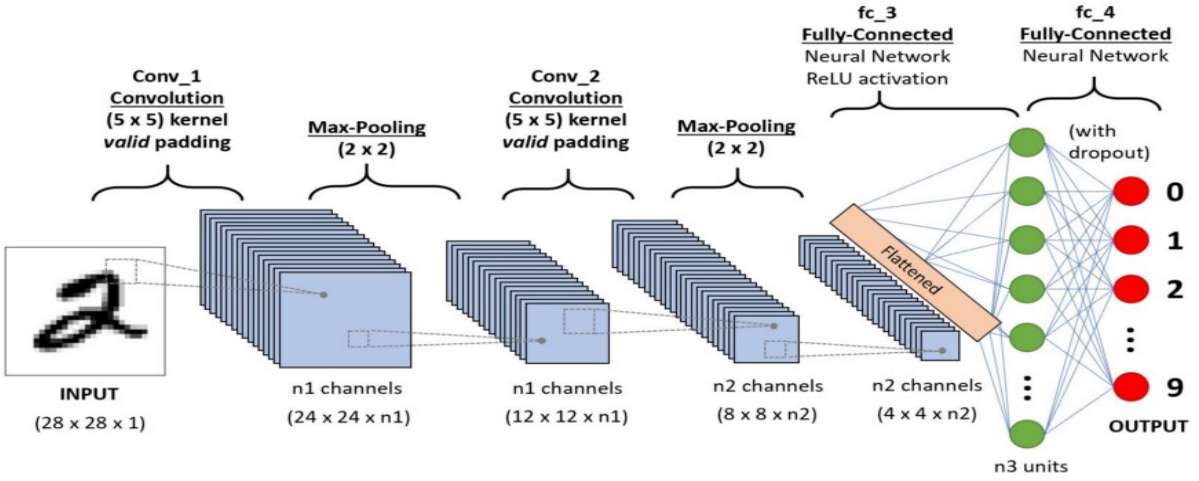
How does CNN work?



CNN is different from other neural networks by its superior performance with image, speech, and audio signal inputs. It has four types of layers, which are:

- Convolutional layer
- Pooling layer
- Flattening
- Fully-connected (FC) layer

Example



Similarities and Differences of ANN vs. CNN

ANN utilizes weights and an activation function for the preponderance of its method. The best way to explain how ANN works is to rebuild how a brain's neural network works artificially. After it gets something incorrect, it "changes" the way it believes, as a human would.

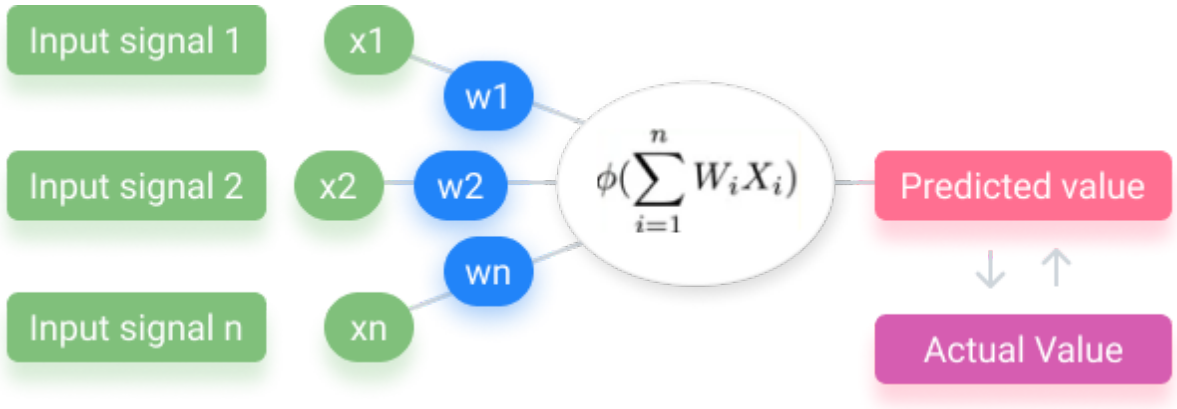
The "layers" in ANN are rows of data ends hosted via neurons that utilize the same neural network. ANN utilizes weights to learn. Weights get changed after each iteration via the neuron in ANN. ANN goes back and changes the weights relying on the accuracy calculated by a "cost function."

Comparatively, there is no neuron or weights in CNN. CNN instead casts numerous layers on images and uses filtration to analyze image inputs. These layers are the math coating, corrected linear unit layer, and fully connected layer. These layers aim to understand patterns that the network can "see," process the output of data, and provide an n-dimensional vector output.

That n-dimensional output follows distinct features and connects them with the image input supplied. It can then give the classification result to the user. Despite their differences, both methods use error measures to improve learning and produce epochs to analyze the effectiveness of models generated.

Input Processing Differences

ANN processes inputs differently than CNN. As a result, ANN is sometimes directed as a Feed-Forward Neural Network because inputs are processed only in a forward-facing direction.



Because of the reliance on accurate data inputs, ANN tends to be a less popular choice when interpreting images. Meanwhile, CNN works in a consistent way with images as input data. We are using filters on image results in feature maps. CNN doesn't process data forward-facing but instead refers to the same data multiple times when creating maps.

ANN and CNN for Image Classification

With ANN must provide concrete data points. For example, in a model where we attempt to differentiate between dogs and cats, the noses' width and the ears' measurement must be explicitly provided as data points.

When using CNN, these spatial features are pulled from image input. This makes CNN ideal when thousands of components need to be pulled. Instead of estimating each feature, CNN gathers these components on its own.

Using ANN, image classification problems become challenging because 2-dimensional images need to be corrected to 1-dimensional vectors. This improves the number of trainable parameters exponentially. Increasing trainable parameters takes repository and processing capacity.

In other words, it would be expensive. Compared to its ancestors, the main advantage of CNN is that it automatically catches the essential features without any human supervision. CNN would be a perfect solution to computer vision and picture classification problems.

ANN vs. CNN for Data Classification

ANN is perfect for solving problems concerning data. Forward-facing algorithms can efficiently process image, text, and tabular data. CNN needs many more data inputs to reach its novel increased accuracy rate.

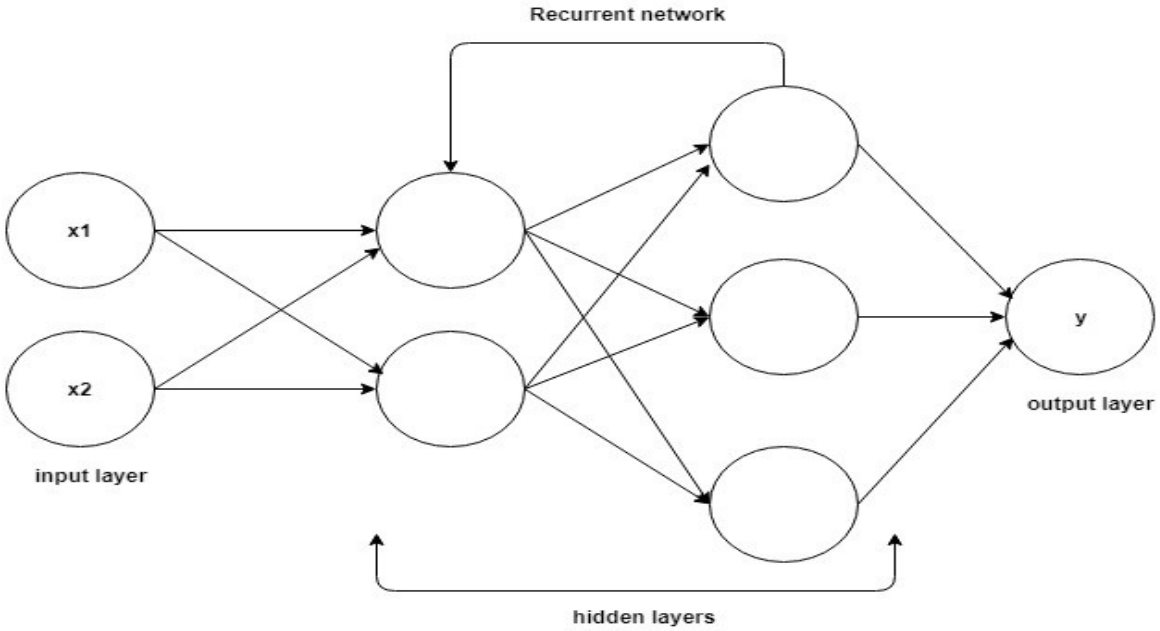
In many cases, to get the identical precision as ANN for data processing, you have to use numerous data augmentation techniques to widen the reaches of your data. In addition, ANN can implicitly notice complex nonlinear associations between dependent and separate variables.

It would also get you virtually the exact precision rate as CNN for data classification problems. This makes CNN an overkill solution for a data classification problem since you'd have to augment your data to widen the dataset and deal with CNN's storage and hardware dependencies. ANN is a comparatively weightless way of solving data classification problems.

Recurrent Neural Network

Working on Neural Network

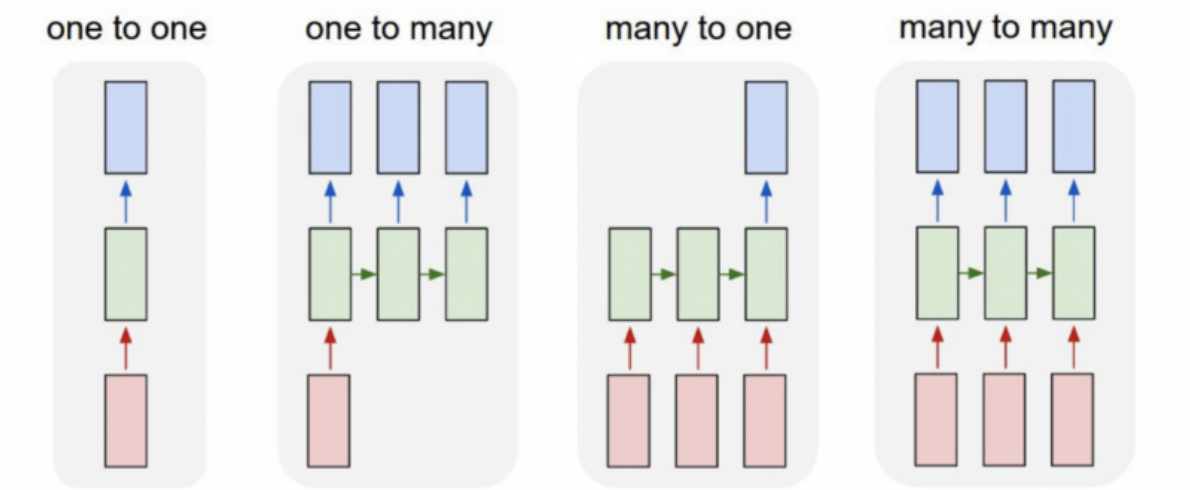
- RNNs are a robust neural network and belong to the most promising algorithms because it is the superior one with interior memory.
- Recurrent Neural Networks(RNN) were originally created in the 1980s, but only in recent years have we seen their true potential. An increase in computational power along with the massive amounts of data that we now have to work with and the invention of long short-term memory (LSTM) in the 1990s, has brought RNNs to the foreground.
- RNN has vast internal memory that helps to remember important information of inputs received, which allows in predicting the next outputs. So preferred algorithm for sequential data like time series, speech, text, financial data, audio, video, weather, and much more. Recurrent neural networks(RNN) can form a much deeper understanding of a sequence and its context compared to other algorithms.
- From the above image let's understand how an RNN works. First, we need to pass an input layer, that will pass through different hidden layers and process the outputs. So what is different in hidden layers of other neural networks and an RNN, because it has two inputs present and recent past that helps to predict the outputs.


Recurrent network
x1
x2
input layer
y
output layer
hidden layers

Type of Recurrent Neural Network

In general, Recurrent Neural Networks are of four types the name reflects the same as types of functions names. They are:

1. One to One
2. One to Many
3. Many to One
4. Many to Many



We use one to many, many to many (translation) and many to one (classifying a voice). Many to many and many to one are most commonly used for voice assistants and most helpful in recurrent neural networks(RNN).

Advantages of Recurrent Neural Network(RNN)

Recurrent Neural Networks is one of the most efficient algorithms. So it has many vast advantages:

- RNN can process the input of any length.
- Even if the input size is larger, the model size does not increase.
- An RNN model is modelled to remember each piece of information throughout the time which is very helpful in any time series predictor.
- The weights can be shared across the time steps.
- RNN can use their internal memory for processing the arbitrary series of inputs which is not the case with feedforward neural networks.

Disadvantages of Recurrent Neural Network(RNN)

- Due to its recurrent nature, the computation is slow.
- Training of RNN models can be difficult.
- If we are using relu or tanh as activation functions, it becomes very difficult to process sequences that are very long.
- Prone to problems such as exploding and gradient vanishing.

Applications of Recurrent Neural Networks(RNN)

RNN is one of the most used efficient algorithms for different applications like:

- Predicting Problems
- Machine Translation
- Speech Recognition
- Language Modelling
- Generation of Text
- Video tagging
- Time Series
- Text Summarization
- Call Centre Analysis
- Generating Image Descriptions

Interview Questions

Q1:- Why do we prefer Convolutional Neural networks (CNN) over Artificial Neural networks (ANN) for image data as input?

1. Feedforward neural networks can learn a single feature presentation of the image. Still, in the case of complex pictures, ANN will fail to give more valuable predictions because it cannot understand pixel dependencies present in the images.

2. CNN can learn multiple layers of characteristic illustrations of an image by involving filters or transformations.

3. In CNN, the number of parameters for the network to learn is hugely lower than the multilayer neural networks since the numeral of units in the network reduces, decreasing the chance of overfitting.

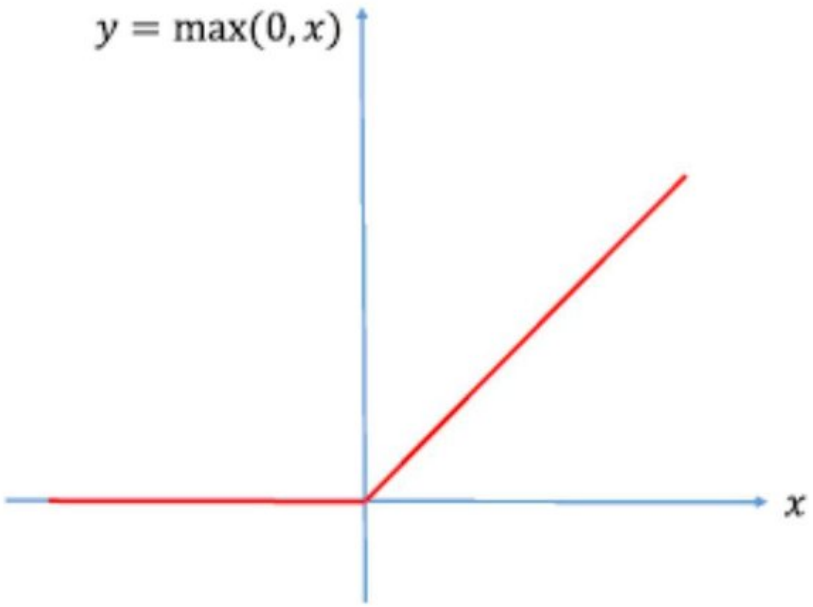
4. Also, CNN considers the context data in the small community, and due to this feature, these are essential to achieve a better forecast in data like images.

Q2:- Explain the significance of the RELU Activation function in Convolution Neural Network.

RELU Layer – Behind each convolution operation, the RELU procedure is used. Moreover, RELU is a non-linear activation function. This process is applied to each pixel and replaces all the negative pixel values in the feature map with zero.

Usually, the image is highly non-linear, which indicates varied pixel values. This is a scenario that is very complicated for an algorithm to make correct predictions. RELU activation function is applied in these cases to reduce the non-linearity and make the job more manageable.

Therefore this layer helps catch elements, reducing the non-linearity of the image and transforming negative pixels to zero, which also permits detecting the variations of features.


$y = \max(0, x)$
x