UNIT-1:-

INTRODUCTION

Human Learning:

- Learning is the act of acquiring new or reinforcing existing knowledge, behaviors, skills or values. Humans have the ability to learn, however with the progress in artificial intelligence, machine learning has become a resource which can augment or even replace human learning. Learning does not happen all at once, but it builds upon and is shaped by previous knowledge. To that end, learning may be viewed as a process, rather than a collection of factual and procedural knowledge.
- Both human as well as machine learning generate knowledge, one residing in the brain the other residing in the machine. This fact raises the question how we apply what kind of knowledge and how we balance these knowledge resources for optimal results. The following discussion hopefully provides some guidance how we can assess this balance keeping in mind that further progress in machine learning and brain research will impact this discussion.

Characteristics of Human Learning:

- Motivation is a important for human learning. Generally one differentiates between intrinsic and extrinsic motivation. Intrinsic motivation is involved when one does something because one enjoys it or finds it interesting, extrinsic motivation when doing something for external rewards or to avoid negative consequences.
- Learning within the context of extrinsic human motivation is closely coupled with formal education and the aim of our teaching institutions to transfer knowledge as efficiently as possible to meet the requirements of employment within various business sectors.
- It provides a **hierarchical model** for the cognitive procedures and goals of learning divided into 6 levels where level 1 is the most basic level for teaching knowledge acquisition and level 6 the top with the highest educational requirements to meet the goals of a specific educational program. Mastering a specific level is a prerequisite to move on to the next higher level. The levels are defined as follows:

1. **Remembering/Memorizing:** defined as the knowing of previously learned material or retrieving, recognizing, and recalling relevant knowledge.
2. **Understanding:** defined as being able to comprehend facts by comparing and interpreting main ideas within the learned material.
3. **Applying:** defined as the ability to use learned material in a new or unprompted way of abstraction and to solve a newly defined problem.
4. **Analyzing:** defined as the ability to examine a problem area and identify the various components (breaking the problem down).
5. **Evaluating:** defined as the ability to make judgments based on criteria or standards or to combine parts to form a new concept or idea.
6. **Creating:** defined as the ability to integrate learning from different areas into a plan for solving a problem and to propose alternative solutions.

Types of Learning:

1. Physical (Kinesthetic) Learning

Physical or kinesthetic learners prefer a hands-on experience rather than listening to lectures or sitting in a class. They like interacting physically with things that are tangible in nature. These learners could see the idea of studying for hours as a daunting experience but are better with actually doing things themselves. They possess qualities like being restless, preferring to get their hands “dirty”, outgoing and energetic.

Ways to engage physical learners:

- Encourage movement within lessons. Example: role play
- Give them well-spaced breaks between lessons to move around
- Use props and interactive models
- Declutter desks to promote better focus

2. Visual (Spatial) Learning

Visual or spatial learners learn best with the help of visual cues like charts, images, diagrams, graphs, etc. These learners respond best to colours and mind maps. They use their visual memory to retain information for longer periods of time. Many visual learners possess characteristics like frequent planning and doodling, they have a good attention span and are extremely observant, and they prefer visual directions.

Ways to engage visual learners:

- Use maps, diagrams, imagery
- Include technology like projectors
- Use colour coding techniques
- Encourage mind maps and flowcharts

3. Auditory Learning

People who tend to understand and retain information by hearing it or saying it out loud (oral) are called auditory learners. These types of learners can quickly notice the change in someone's pitch, tone, and other voice qualities. They usually prefer discussing topics, participating in debates, and conversing about things to remember them. Most auditory learners are easy to distract and might even hum, sing, or talk to self frequently.

Ways to engage auditory learners:

- Try using different pitches and tones while reading the material
- Record voice lessons
- Encourage class presentations, group discussions, debates
- Ask them to teach others verbally

4. Verbal (Read/Write) Learning

These types of learners prefer traditional methods like using multiple written resources for learning. Verbal learners learn best through written material or by writing the material themselves. They usually possess a broad vocabulary and might even like using tools like acronyms, rhymes, tongue twisters, among others. Verbal learners are known to be bookworms.

Ways to engage verbal learners:

- Make use of mnemonics while teaching (song, rhyme, acronym, phrase)
- Inculcate scripts
- Encourage students to jot down and voice their ideas
- Include word games like crossword

5. Logical (Mathematical) Learning

Logical or mathematical learners tend to categorize information into groups to learn them better. They have a knack for quickly recognizing patterns and sequences; and understand equations, numbers, and relationships easily. These learners love structure and logic to things. Naturally, mathematics comes easy to them.

Ways to engage logical learners:

- Create an easy to navigate system to your lessons
- Try and inculcate statistics to subjects other than mathematics
- Classify concepts into groups or categories
- Generate cause-effect relationships between variables throughout all subject areas

6. Musical Learning

Where music or background noise is a distraction to most of us, musical learners prefer them. They tend to learn better with music, beats, and rhythm. Like logical learners, they too find patterns and relationships, but between different sounds. Some sources say they even think in sounds and rhythms instead of words and pictures.

Clearly, these learners often grow up to be musicians or instrumentalists. More often than never, some people are a combination of auditory and musical learners. This is why strategies to engage these two kinds aren't too different.

Ways to engage musical learners:

- Encourage listening to soft background music
- Promote podcasts

7. Naturalist Learners

Naturalist learners learn best through experimentation and practical experiences. They like making observations of the world around them. Just like the name suggests, naturalistic learners are also said to be one with nature. They retain information best when they are outdoors, around plants, animals, among others.

These types can also be somewhat related to kinesthetic learners since they appreciate tactile sensations. All-in-all, they apply scientific reasoning to the world around them and are highly interested in nature, as well as the things created by man.

Ways to engage naturalistic learners:

- Take students out for a field trip
- Give lessons in outdoor spaces
- Promote journaling, drawing, sketching, photographing or natural phenomena
- Encourage work that involves getting into nature (especially in subjects like biology)

8. Linguistic Learners

Linguistic learners are the combination of auditory and verbal learners. They absorb knowledge best by writing, reading, and sounding the material out. These learners can use the traditional methods of learning just like verbal learners and also prefer listening to the information. Linguistic learners also make their own notes while studying.

Ways to engage linguistic learners:

- Read out to them and have them read it back to you
- Include written projects and assignments
- Avoid using too many diagrams; use verbal methods of engaging them

- Avoid using a monotonous voice; use different pitches, voices, and characters

9. Interpersonal (Social) Learners

Social or interpersonal learners learn best while working in groups or with other people. They often make good leaders and others even come for advice to them. Social learners learn by relating their ideas and thoughts to the lives of other people. These learners are usually empaths and possess qualities like sensitivity to others, excellent communication, leadership skills, and problem-solving skills. This type of learning can fall adjacent to one or more types mentioned above.

Ways to engage social learners:

- Figure out their adjacent learning style and inculcate those strategies
- Encourage role-playing
- Assign group activities and projects

10. Intrapersonal (Solitary) Learners

In a complete contrast to interpersonal or social learners, intrapersonal or solitary learners prefer solitude while studying. They are more independent and introspective by nature and prefer to be with their own thoughts and ideas without too much external interference. Usually, you can find these types sitting at the back of the class or you might refer to them as the “quiet kid” but they may end up acing the exam. Solitary learning too can fall adjacent with other learning styles.

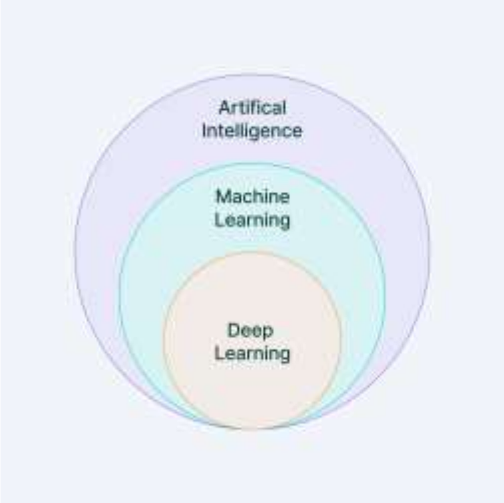
Ways to engage solitary learners:

- Figure out their adjacent learning style and inculcate those strategies
- Designate a quiet area
- Check in with them every once in a while
- Define a specific time for collaboration so they feel prepared enough

Machine learning:

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Machine learning is used in many different applications, from image and speech recognition to natural language processing, recommendation systems, fraud detection, portfolio optimization, automated task, and so on. Machine learning models are also used to power autonomous vehicles, drones, and robots, making them more intelligent and adaptable to changing environments.



How machine learning algorithms work

Machine Learning works in the following manner.

- **Forward Pass:** In the Forward Pass, the machine learning algorithm takes in input data and produces an output. Depending on the model algorithm it computes the predictions.
- **Loss Function:** The loss function, also known as the error or cost function, is used to evaluate the accuracy of the predictions made by the model. The function compares the predicted output of the model to the actual output and calculates the difference between them. This difference is known as error or loss. The goal of the model is to minimize the error or loss function by adjusting its internal parameters.
- **Model Optimization Process:** The model optimization process is the iterative process of adjusting the internal parameters of the model to minimize the error or loss function. This is done using an optimization algorithm, such as **gradient descent**. The optimization algorithm calculates the gradient of the error function with respect to the model's parameters and uses this information to adjust the parameters to reduce the error. The algorithm repeats this process until the error is minimized to a satisfactory level.

Once the model has been trained and optimized on the training data, it can be used to make predictions on new, unseen data. The accuracy of the model's predictions can be evaluated using various performance metrics, such as accuracy, precision, recall, and F1-score.

Machine Learning lifecycle:

The lifecycle of a machine learning project involves a series of steps that include:

1. **Study the Problems:** The first step is to study the problem. This step involves understanding the business problem and defining the objectives of the model.
2. **Data Collection:** When the problem is well-defined, we can collect the relevant data required for the model. The data could come from various sources such as databases, APIs, or web scraping.
3. **Data Preparation:** When our problem-related data is collected. then it is a good idea to check the data properly and make it in the desired format so that it can be used by the model to find the hidden patterns. This can be done in the following steps:

- Data cleaning
- Data Transformation
- Explanatory Data Analysis and Feature Engineering
- Split the dataset for training and testing.

4. **Model Selection:** The next step is to select the appropriate machine learning algorithm that is suitable for our problem. This step requires knowledge of the strengths and weaknesses of different algorithms. Sometimes we use multiple models and compare their results and select the best model as per our requirements.
5. **Model building and Training:** After selecting the algorithm, we have to build the model.
   1. In the case of traditional machine learning building mode is easy it is just a few hyperparameter tunings.
   2. In the case of deep learning, we have to define layer-wise architecture along with input and output size, number of nodes in each layer, loss function, gradient descent optimizer, etc.
   3. After that model is trained using the preprocessed dataset.
6. **Model Evaluation:** Once the model is trained, it can be evaluated on the test dataset to determine its accuracy and performance using different techniques like classification report, F1 score, precision, recall, ROC Curve, Mean Square error, absolute error, etc.
7. **Model Tuning:** Based on the evaluation results, the model may need to be tuned or optimized to improve its performance. This involves tweaking the hyperparameters of the model.
8. **Deployment:** Once the model is trained and tuned, it can be deployed in a production environment to make predictions on new data. This step requires integrating the model into an existing software system or creating a new system for the model.
9. **Monitoring and Maintenance:** Finally, it is essential to monitor the model's performance in the production environment and perform maintenance tasks as required. This involves monitoring for data drift, retraining the model as needed, and updating the model as new data becomes available.

Various Applications of Machine Learning

Now in this Machine learning tutorial, let's learn the applications of Machine Learning:

- **Automation**: Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots perform the essential process steps in manufacturing plants.
- **Finance Industry**: Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.
- **Government organization**: The government makes use of ML to manage public safety and utilities. Take the example of China with its massive face recognition. The government uses Artificial intelligence to prevent jaywalking.
- **Healthcare industry**: Healthcare was one of the first industries to use machine learning with image detection.
- **Marketing:** Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, the

marketing department relies on AI to optimize customer relationships and marketing campaigns.
- **Retail industry**: Machine learning is used in the retail industry to analyze customer behavior, predict demand, and manage inventory. It also helps retailers to personalize the shopping experience for each customer by recommending products based on their past purchases and preferences.
- **Transportation**: Machine learning is used in the transportation industry to optimize routes, reduce fuel consumption, and improve the overall efficiency of transportation systems. It also plays a role in autonomous vehicles, where ML algorithms are used to make decisions about navigation and safety.

The Advantages of Machine Learning

The wide application of ML speaks in favor of its clear advantages:

- **ML is efficient in pattern-finding.** Machine learning, in a short time, notices patterns and repetitions in massive data sets, which would take humans significantly more time, and even then, it is not sure that they would notice the patterns.

- **ML work is automated** and does not require human intervention. Thanks to ML, you no longer have to supervise your project at every stage. Giving computers the ability to learn enables them to make predictions and enhance algorithms.

- **ML algorithms are ever-evolving.** Machine learning algorithms become more accurate as they gain experience. They can consequently make wiser selections. Algorithms become faster at making more accurate predictions as the data set expands.

- **ML can work with different types of data** – labeled or unlabeled, visual or textual, ML algorithms can handle them.

- **Wide application.** Various industries can use ML algorithms in multiple sectors: marketing, eCommerce, finance, healthcare, agriculture, and many others. Regardless of the industry in which they are applied, ML algorithms enable better targeting and recommendations, which improves the entire user experience.

The Disadvantages of Machine Learning

Machine learning is an advanced technology, however, it has certain disadvantages:

- **ML is limited by the amount of information obtained**: Machine learning is not based on human knowledge, which is inexhaustible, but on the data it receives from data scientists. Therefore, ML is still far from mimicking human intelligence.

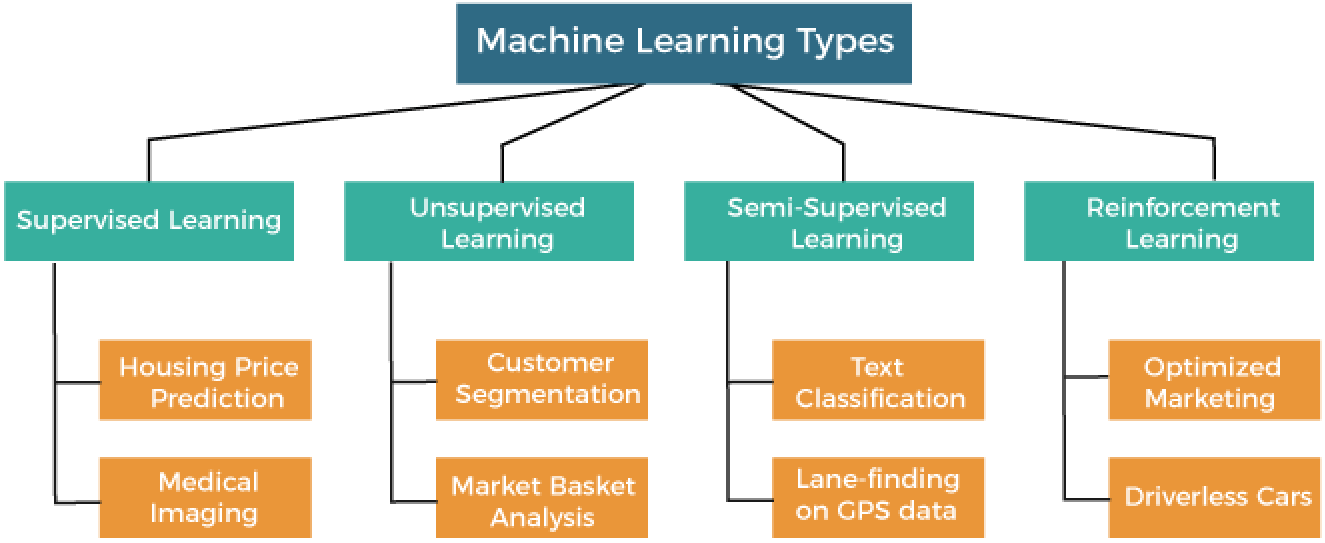
- **ML algorithms are difficult to train:** the process is time-consuming, expensive, and requires large databases. Only **4%** of data training scientists report that they didn't have difficulties during this process. Problems occur due to data errors, unusable data formats, and the inability to label data.

- **ML exact working methods are difficult to identify:** Machine learning systems work independently, making decisions, and if you notice irregular work, there is no way to know why it happened. All you can do is start the process from the beginning, training the algorithms with a new set of data.

Types of machine learning:

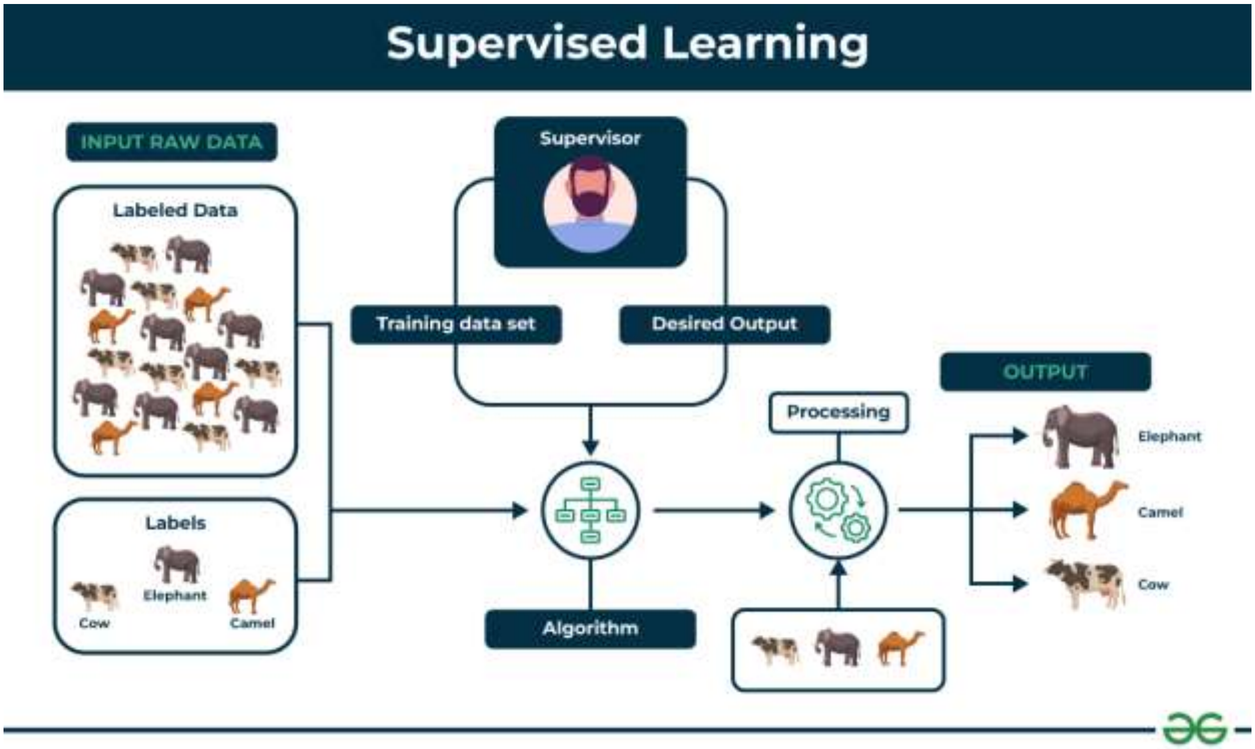
Based on the methods and way of learning, machine learning is divided into mainly four types, which are:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Learning



1. Supervised Machine Learning:

Supervised learning is defined as when a model gets trained on a **“Labelled Dataset”**. Labelled datasets have both input and output parameters. In **Supervised Learning** algorithms learn to map points between inputs and correct outputs. It has both training and validation datasets labelled.



Example: Consider a scenario where you have to build an image classifier to differentiate between cats and dogs. If you feed the datasets of dogs and cats labelled images to the algorithm, the machine will learn to classify between a dog or a cat from these labeled images. When we input new dog or cat images that it has never seen before, it will use the learned algorithms and predict whether it is a dog or a cat. This is how **supervised learning** works, and this is particularly an image classification.

Categories of Supervised Machine Learning

Supervised machine learning can be classified into two types of problems, which are given below:

- **Classification**
- **Regression**

a) Classification

Classification algorithms are used to solve the classification problems in which the output variable is categorical, such as "**Yes" or No, Male or Female, Red or Blue, etc**. The classification algorithms predict the categories present in the dataset. Some real-world examples of classification algorithms are **Spam Detection, Email filtering, etc.**

Some popular classification algorithms are given below:

- **Random Forest Algorithm**
- **Decision Tree Algorithm**
- **Logistic Regression Algorithm**
- **Support Vector Machine Algorithm**

b) Regression

Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables. These are used to predict continuous output variables, such as market trends, weather prediction, etc.

Some popular Regression algorithms are given below:

- **Simple Linear Regression Algorithm**
- **Multivariate Regression Algorithm**
- **Decision Tree Algorithm**
- **Lasso Regression**

Applications of Supervised Learning

Supervised learning is used in a wide variety of applications, including:

- **Image classification**: Identify objects, faces, and other features in images.
- **Natural language processing:** Extract information from text, such as sentiment, entities, and relationships.
- **Speech recognition**: Convert spoken language into text.
- **Recommendation systems**: Make personalized recommendations to users.
- **Predictive analytics**: Predict outcomes, such as sales, customer churn, and stock prices.
- **Medical diagnosis**: Detect diseases and other medical conditions.
- **Fraud detection**: Identify fraudulent transactions.
- **Autonomous vehicles**: Recognize and respond to objects in the environment.
- **Email spam detection**: Classify emails as spam or not spam.
- **Quality control in manufacturing**: Inspect products for defects.
- **Credit scoring**: Assess the risk of a borrower defaulting on a loan.
- **Gaming**: Recognize characters, analyze player behavior, and create NPCs.

- **Customer support**: Automate customer support tasks.
- **Weather forecasting**: Make predictions for temperature, precipitation, and other meteorological parameters.
- **Sports analytics**: Analyze player performance, make game predictions, and optimize strategies.

Advantages of Supervised Machine Learning

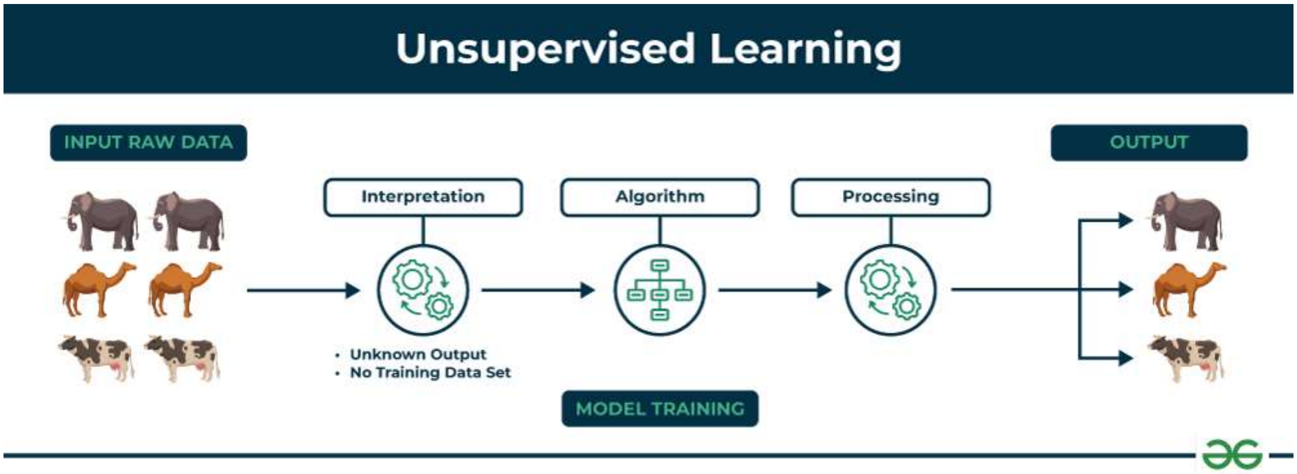
- **Supervised Learning** models can have high accuracy as they are trained on **labelled data**.
- The process of decision-making in supervised learning models is often interpretable.
- It can often be used in pre-trained models which saves time and resources when developing new models from scratch.

Disadvantages of Supervised Machine Learning

- It has limitations in knowing patterns and may struggle with unseen or unexpected patterns that are not present in the training data.
- It can be time-consuming and costly as it relies on **labeled** data only.
- It may lead to poor generalizations based on new data.

2. Unsupervised Machine Learning:

Unsupervised Learning Unsupervised learning is a type of machine learning technique in which an algorithm discovers patterns and relationships using unlabeled data. Unlike supervised learning, unsupervised learning doesn’t involve providing the algorithm with labeled target outputs. The primary goal of Unsupervised learning is often to discover hidden patterns, similarities, or clusters within the data, which can then be used for various purposes, such as data exploration, visualization, dimensionality reduction, and more.



Unsupervised Learning

Example: Consider that you have a dataset that contains information about the purchases you made from the shop. Through clustering, the algorithm can group the same purchasing behavior among you and other customers, which reveals potential customers without predefined labels. This type of information can help businesses get target customers as well as identify outliers.

Categories of Unsupervised Machine Learning

Unsupervised Learning can be further classified into two types, which are given below:

- **Clustering**
- **Association**

1) Clustering

The clustering technique is used when we want to find the inherent groups from the data. It is a way to group the objects into a cluster such that the objects with the most similarities remain in one group and have fewer or no similarities with the objects of other groups. An example of the clustering algorithm is grouping the customers by their purchasing behaviour.

Some of the popular clustering algorithms are given below:

- **K-Means Clustering algorithm**
- **Mean-shift algorithm**
- **DBSCAN Algorithm**
- **Principal Component Analysis**
- **Independent Component Analysis**

2) Association

Association rule learning is an unsupervised learning technique, which finds interesting relations among variables within a large dataset. The main aim of this learning algorithm is to find the dependency of one data item on another data item and map those variables accordingly so that it can generate maximum profit. This algorithm is mainly applied in **Market Basket analysis, Web usage mining, continuous production**, etc.

Some popular algorithms of Association rule learning are **Apriori Algorithm, Eclat, FP-growth algorithm.**

Applications of Unsupervised Learning

Here are some common applications of unsupervised learning:

- **Clustering**: Group similar data points into clusters.
- **Anomaly detection**: Identify outliers or anomalies in data.
- **Dimensionality reduction**: Reduce the dimensionality of data while preserving its essential information.

- **Recommendation systems**: Suggest products, movies, or content to users based on their historical behavior or preferences.
- **Topic modeling**: Discover latent topics within a collection of documents.
- **Density estimation**: Estimate the probability density function of data.
- **Image and video compression**: Reduce the amount of storage required for multimedia content.
- **Data preprocessing**: Help with data preprocessing tasks such as data cleaning, imputation of missing values, and data scaling.
- **Market basket analysis**: Discover associations between products.
- **Genomic data analysis**: Identify patterns or group genes with similar expression profiles.
- **Image segmentation**: Segment images into meaningful regions.
- **Community detection in social networks**: Identify communities or groups of individuals with similar interests or connections.
- **Customer behavior analysis**: Uncover patterns and insights for better marketing and product recommendations.
- **Content recommendation**: Classify and tag content to make it easier to recommend similar items to users.
- **Exploratory data analysis (EDA)**: Explore data and gain insights before defining specific tasks.

Advantages of Unsupervised Machine Learning

- It helps to discover hidden patterns and various relationships between the data.
- Used for tasks such as **customer segmentation, anomaly detection,** and **data exploration**.
- It does not require labeled data and reduces the effort of data labeling.

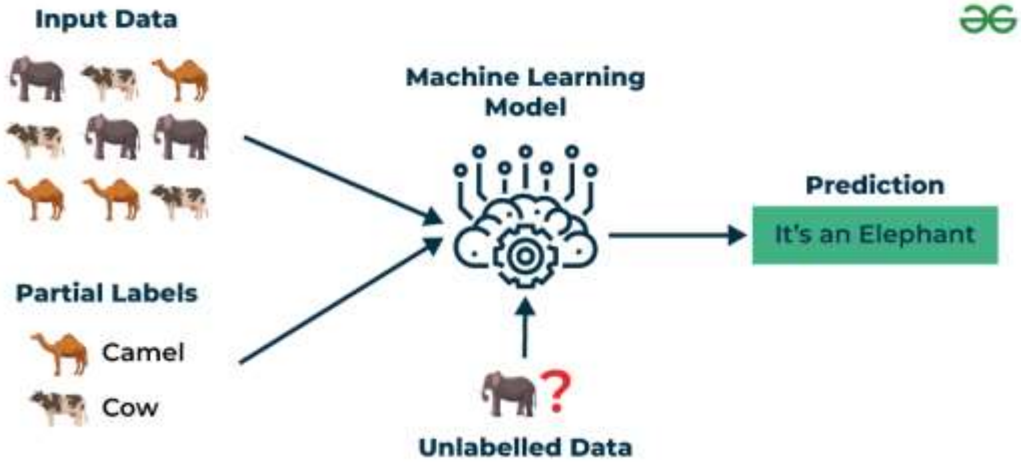
Disadvantages of Unsupervised Machine Learning

- Without using labels, it may be difficult to predict the quality of the model's output.
- Cluster Interpretability may not be clear and may not have meaningful interpretations.
- It has techniques such as autoencoders and dimensionality reduction that can be used to extract meaningful features from raw data.

3. Semi-Supervised Learning:

Semi-Supervised learning is a machine learning algorithm that works between the supervised and unsupervised learning so it uses both **labelled and unlabelled** data. It's particularly useful when obtaining labeled data is costly, time-consuming, or resource-intensive. This approach is useful when the dataset is expensive and time-consuming. Semi-supervised learning is chosen when labeled data requires skills and relevant resources in order to train or learn from it.

We use these techniques when we are dealing with data that is a little bit labeled and the rest large portion of it is unlabeled. We can use the unsupervised techniques to predict labels and then feed these labels to supervised techniques. This technique is mostly applicable in the case of image data sets where usually all images are not labeled.



Semi-Supervised Learning

Types of Semi-Supervised Learning Methods

There are a number of different semi-supervised learning methods each with its own characteristics. Some of the most common ones include:

- **Graph-based semi-supervised learning:** This approach uses a graph to represent the relationships between the data points. The graph is then used to propagate labels from the labeled data points to the unlabeled data points.
- **Label propagation:** This approach iteratively propagates labels from the labeled data points to the unlabeled data points, based on the similarities between the data points.
- **Co-training:** This approach trains two different machine learning models on different subsets of the unlabeled data. The two models are then used to label each other’s predictions.
- **Self-training:** This approach trains a machine learning model on the labeled data and then uses the model to predict labels for the unlabeled data. The model is then retrained on the labeled data and the predicted labels for the unlabeled data.
- **Generative adversarial networks (GANs):** GANs are a type of deep learning algorithm that can be used to generate synthetic data. GANs can be used to generate unlabeled data for semi-supervised learning by training two neural networks, a generator and a discriminator.

Example: Consider that we are building a language translation model, having labeled translations for every sentence pair can be resources intensive. It allows the models to learn from labeled and unlabeled sentence pairs, making them more accurate. This technique has led to significant improvements in the quality of machine translation services.

Applications of Semi-Supervised Learning

Here are some common applications of semi-supervised learning:

- **Image Classification and Object Recognition**: Improve the accuracy of models by combining a small set of labeled images with a larger set of unlabeled images.

- **Natural Language Processing (NLP)**: Enhance the performance of language models and classifiers by combining a small set of labeled text data with a vast amount of unlabeled text.
- **Speech Recognition:** Improve the accuracy of speech recognition by leveraging a limited amount of transcribed speech data and a more extensive set of unlabeled audio.
- **Recommendation Systems**: Improve the accuracy of personalized recommendations by supplementing a sparse set of user-item interactions (labeled data) with a wealth of unlabeled user behavior data.
- **Healthcare and Medical Imaging**: Enhance medical image analysis by utilizing a small set of labeled medical images alongside a larger set of unlabeled images.

Advantages of Semi- Supervised Machine Learning

- It leads to better generalization as compared to **supervised learning,** as it takes both labeled and unlabeled data.
- Can be applied to a wide range of data.

Disadvantages of Semi- Supervised Machine Learning

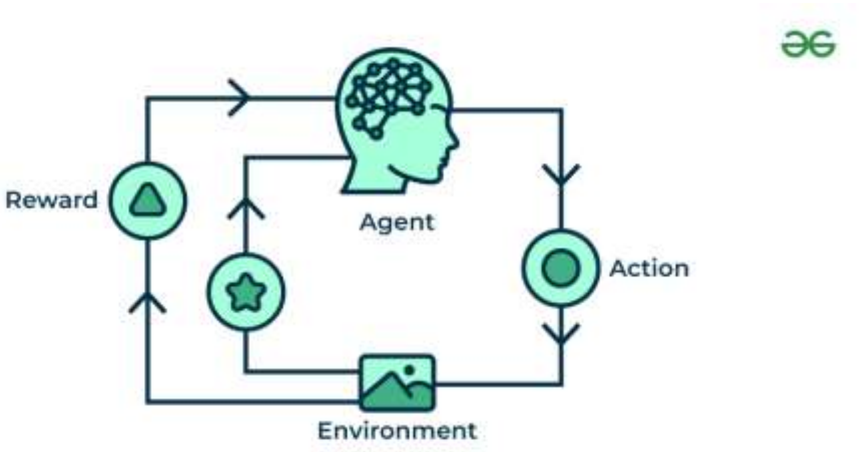
- **Semi-supervised** methods can be more complex to implement compared to other approaches.
- It still requires some **labeled data** that might not always be available or easy to obtain.
- The unlabeled data can impact the model performance accordingly.

4. Reinforcement Machine Learning:

Reinforcement machine learning algorithm is a learning method that interacts with the environment by producing actions and discovering errors. **Trial, error, and delay** are the most relevant characteristics of reinforcement learning. In this technique, the model keeps on increasing its performance using Reward Feedback to learn the behavior or pattern. These algorithms are specific to a particular problem e.g. Google Self Driving car, AlphaGo where a bot competes with humans and even itself to get better and better performers in Go Game. Each time we feed in data, they learn and add the data to their knowledge which is training data. So, the more it learns the better it gets trained and hence experienced.

Here are some of most common reinforcement learning algorithms:

- **Q-learning:** Q-learning is a model-free RL algorithm that learns a Q-function, which maps states to actions. The Q-function estimates the expected reward of taking a particular action in a given state.
- **SARSA (State-Action-Reward-State-Action):** SARSA is another model-free RL algorithm that learns a Q-function. However, unlike Q-learning, SARSA updates the Q-function for the action that was actually taken, rather than the optimal action.
- **Deep Q-learning:** Deep Q-learning is a combination of Q-learning and deep learning. Deep Q-learning uses a neural network to represent the Q-function, which allows it to learn complex relationships between states and actions.



Reinforcement Machine Learning

Example: Consider that you are training an AI agent to play a game like chess. The agent explores different moves and receives positive or negative feedback based on the outcome. Reinforcement Learning also finds applications in which they learn to perform tasks by interacting with their surroundings.

Categories of Reinforcement Learning

Reinforcement learning is categorized mainly into two types of methods/algorithms:

- **Positive Reinforcement Learning:** Positive reinforcement learning specifies increasing the tendency that the required behaviour would occur again by adding something. It enhances the strength of the behaviour of the agent and positively impacts it.
- **Negative Reinforcement Learning:** Negative reinforcement learning works exactly opposite to the positive RL. It increases the tendency that the specific behaviour would occur again by avoiding the negative condition.

Applications of Reinforcement Machine Learning

Here are some applications of reinforcement learning:

- **Game Playing**: RL can teach agents to play games, even complex ones.
- **Robotics**: RL can teach robots to perform tasks autonomously.
- **Autonomous Vehicles**: RL can help self-driving cars navigate and make decisions.
- **Recommendation Systems**: RL can enhance recommendation algorithms by learning user preferences.
- **Healthcare**: RL can be used to optimize treatment plans and drug discovery.
- **Natural Language Processing (NLP)**: RL can be used in dialogue systems and chatbots.
- **Finance and Trading**: RL can be used for algorithmic trading.

- **Supply Chain and Inventory Management**: RL can be used to optimize supply chain operations.
- **Energy Management**: RL can be used to optimize energy consumption.
- **Game AI**: RL can be used to create more intelligent and adaptive NPCs in video games.
- **Adaptive Personal Assistants**: RL can be used to improve personal assistants.
- **Virtual Reality (VR) and Augmented Reality (AR):** RL can be used to create immersive and interactive experiences.
- **Industrial Control**: RL can be used to optimize industrial processes.
- **Education**: RL can be used to create adaptive learning systems.
- **Agriculture**: RL can be used to optimize agricultural operations.

Advantages of Reinforcement Machine Learning

- It has autonomous decision-making that is well-suited for tasks and that can learn to make a sequence of decisions, like robotics and game-playing.
- This technique is preferred to achieve long-term results that are very difficult to achieve.
- It is used to solve a complex problems that cannot be solved by conventional techniques.

Disadvantages of Reinforcement Machine Learning

- Training Reinforcement Learning agents can be computationally expensive and time-consuming.
- Reinforcement learning is not preferable to solving simple problems.
- It needs a lot of data and a lot of computation, which makes it impractical and costly.

Problems not to be solved:

Machine learning (ML) is an integral part of data science and has played a vital role in the growth of various industries. However, despite its advantages, there are several unresolved issues and challenges faced by machine learning professionals. This article aims to explore some of the significant challenges in machine learning, including overfitting, explainability, fairness, privacy, and interpretability.

Overfitting

Overfitting is a common problem in machine learning, where the model performs well on the training data but poorly on the testing data. To address this issue, machine learning professionals use regularization techniques like L1 or L2 regularization or use more data for training.

Explainability

Explainability is the ability to understand how a model makes predictions, and it is an essential requirement for machine learning models. The need for explainability is particularly important in critical applications such as healthcare and finance. Understanding how a model makes decisions is crucial to ensure that the model is transparent, ethical, and reliable.

Fairness

Fairness is a critical challenge in machine learning, where models can discriminate against certain groups or individuals. To address fairness issues, techniques such as demographic parity, equal opportunity, or equalized odds can be used. Fairness is essential to ensure that machine learning models do not perpetuate existing biases and discriminate against specific groups or individuals.

Privacy

Privacy is another significant challenge in machine learning, where models may inadvertently reveal personal information about individuals. Techniques such as differential privacy or federated learning can be used to ensure that models do not compromise personal privacy. Maintaining privacy is essential to ensure that machine learning models do not violate ethical and legal obligations related to personal information.

Machine learning professionals must take steps to ensure that datasets are properly anonymized, and privacy techniques are implemented to protect personal information. Failure to address privacy concerns in machine learning can have significant consequences and may lead to legal or ethical violations.

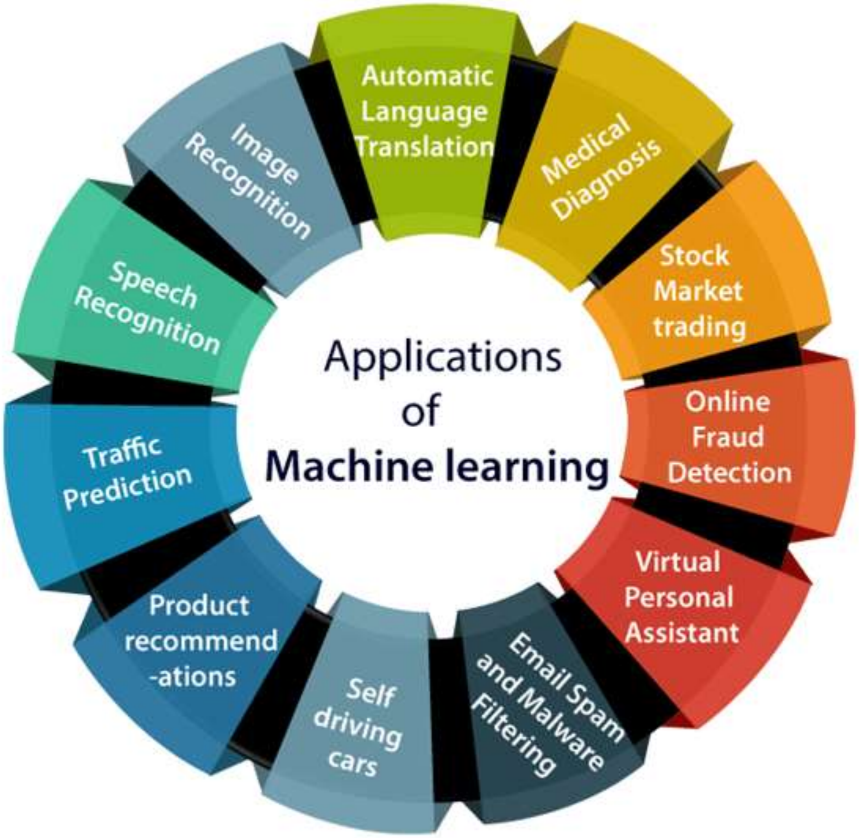
Interpretability

Interpretability is the ability to understand the underlying factors that contribute to a model's predictions. It is particularly important in applications such as medical diagnosis, where it is essential to understand the factors that contribute to a particular diagnosis. Interpretability is crucial to ensure that machine learning models are transparent, reliable, and ethical.

Applications:

Applications of Machine learning:

Machine learning is a buzzword for today's technology, and it is growing very rapidly day by day. We are using machine learning in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Below are some most trending real-world applications of Machine Learning:


Applications of Machine learning
Automatic Language Translation
Medical Diagnosis
Stock Market trading
Online Fraud Detection
Virtual Personal Assistant
Email Spam and Malware Filtering
Self driving cars
Product recommend -ations
Traffic Prediction
Speech Recognition
Image Recognition

1. Image Recognition:

Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, **Automatic friend tagging suggestion**:

Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's **face detection** and **recognition algorithm**.

It is based on the Facebook project named "**Deep Face**," which is responsible for face recognition and person identification in the picture.

2. Speech Recognition:

While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning.

Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**." At present, machine learning algorithms are widely used by various applications of speech recognition. **Google assistant**, **Siri**, **Cortana**, and **Alexa** are using speech recognition technology to follow the voice instructions.

3. Traffic prediction:

If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.

It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:

- **Real Time location** of the vehicle form Google Map app and sensors
- **Average time has taken** on past days at the same time.

Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

4. Product recommendations:

Machine learning is widely used by various e-commerce and entertainment companies such as **Amazon**, **Netflix**, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.

Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest.

As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

5. Self-driving cars:

One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

6. Email Spam and Malware Filtering:

Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Below are some spam filters used by Gmail:

- Content Filter
- Header filter

- General blacklists filter
- Rules-based filters
- Permission filters

Some machine learning algorithms such as **Multi-Layer Perceptron**, **Decision tree**, and **Naïve Bayes classifier** are used for email spam filtering and malware detection.

7. Virtual Personal Assistant:

We have various virtual personal assistants such as **Google assistant**, **Alexa**, **Cortana**, **Siri**. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.

These virtual assistants use machine learning algorithms as an important part.

These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

8. Online Fraud Detection:

Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as **fake accounts**, **fake ids**, and **steal money** in the middle of a transaction. So to detect this, **Feed Forward Neural network** helps us by checking whether it is a genuine transaction or a fraud transaction.

For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.

9. Stock Market trading:

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning's **long short term memory neural network** is used for the prediction of stock market trends.

10. Medical Diagnosis:

In medical science, machine learning is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain.

It helps in finding brain tumors and other brain-related diseases easily.

11. Automatic Language Translation:

Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.

The technology behind the automatic translation is a sequence to sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.

Languages/Tools–

Languages:

1. Python:

Python leads all the other languages with more than 60% of machine learning developers are using and prioritizing it for development because python is easy to learn. Scalable and open source. Python has many awesome visualization packages and useful core libraries like Numpy, scipy, pandas, matplotlib, seaborn, sklearn which really makes your work very easy and empower the machines to learn.

- **Numpy:** Numeric Python or Numpy is a Linear Algebra Library for Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices.
- **Pandas:** It is the most popular Python library which provides highly optimized performance for data analysis.
- **Matplotlib:** It is a popular python plotting library used for creating basic graphs like line charts, bar charts, histograms, and many more.
- **Seaborn:** Provides a high-level interface for creating attractive graphs
- **sci-kit Learn:** It is used for data mining and data analysis which implements a wide-range of machine-learning algorithms like classification, regression, and clustering algorithms including support vector machines, random forests, gradient boosting, k-means.

2. Java:

This programming language is the "Jack of all the trade" and continues to dominate over in the ML industry also. Java provides many good environments like Weka, Knime, RapidMiner, Elka which used to perform machine learning tasks using graphical user interfaces.

- **Weka:** It is a free, portable library primarily used for data mining, data analysis, and predictive modelling and best used for machine learning algorithms. it is easy to use with the graphical interface and supports several standard data mining tasks, including data preprocessing, classification, clustering, and feature selection.

- **JavaML:** A Java API with simple and easy interfaces to implement the collection of machine learning and data mining algorithms in Java with clearly written and properly documented implementation of algorithms.
- **Deeplearning4j:** It is an innovative open-source distributed deep learning library that provides a computing framework with wide support for machine learning algorithms. This library is extremely useful for identifying patterns, sentiment, sound, and text and is designed especially for business environments.
- **ELKI:** It is a unique open-source data mining framework that mainly focused in the independent evaluation of data mining algorithms and data management and emphasizes in unsupervised methods. It also allows arbitrary data types, file formats, or distance or similarity measures.

3. C++:

The superfast C++ programming language is also very popular in the field of machine learning. This powerful language gets supported by most of the machine learning platforms. If you have some good working knowledge using C++ then it is a pretty good idea to learn machine learning using C++. C++ is much efficient compare to most of programming languages. Many powerful libraries such as TensorFlow and Torch are implemented in the C++ programming language so machine learning and C++ is truly a great combination.

- **TensorFlow:** Google's open-source TensorFlow is used to do numerical computations on any CPU or GPU using data flow graphs and make decisions with whatever information it gets.
- **Torch:** A open-source machine learning library which makes scientific and numerical operation easier by providing a large number of algorithms. it makes for easier and improved efficiency and speed.
- **mlpack:** A superfast, flexible machine learning library which provides fast and extensible implementations of cutting-edge machine learning algorithms using C++ classes which can be integrated into larger-scale machine learning solutions

4. R:

R is a very popular programming language for statistical computing, analysis, and visualizations in machine learning. It is a perfect graphics-based language for exploring the statistical data via graph vastly used by data professionals at Facebook, Google, etc. Though R is highly preferable in bioengineering and biomedical statistic it is also popular in implementing machine learning like regression, classification, and decision tree formation.

- **xgboost:** this is used for implementing the gradient boosting framework and is popular for it's performance and speed. It supports various objective functions like regression, classification, and ranking and is extensible so that you can define your own objectives easily.
- **mlr:** It is an extensible framework for classification, regression, and clustering problems and has easy extension mechanism through s3 inheritance.
- **PARTY:** this package is used for recursive partitioning. This package is used to build decision trees based on the Conditional Inference algorithm. This package is also extensive, which reduces the training time and bias.

- **CARET:** this package is developed to combine model training and prediction for several different algorithms for a given business problem and helps to choose the best machine learning algorithm.

5. Javascript:

It is one of the most widely used, high-level, and dynamically typed languages which is flexible and multi-paradigm. Javascript is also so popular in ML that high-profile projects like Google's Tensorflow.js are based on JavaScript. If you are a master of Javascript then literally you can do everything from full-stack to machine learning and NLP.

- **Brain.js:** It is a GPU accelerated, easy to integrate neural networks in JavaScript which is used with Node.js in the browser and provides multiple neural network implementations to train to do different things well. It is so simple to use that you do not need to know Neural Networks in detail to work with this.
- **Tensorflow.js:** It is a popular library for machine learning in JavaScript. You can build and train models directly in JavaScript using flexible APIs and almost any problems in Machine Learning can be solved using Tensorflow.js. You can also retrain the existing ML models using your own data.
- **machinelearn.js:** It is the savior of Javascript which is a replacement of python's ScikitLearn library. It provides clustering, decomposition, feature extractions models, and utilities for supervised and unsupervised learning.
- **face-api.js:** A ready-to-use APIs that includes implementations of well-known Face Detection and Recognition models which is pre-trained with a wide variety of datasets. It gives you the flexibility to directly plug into any Node.js and browser environments. Being lightweight this library can be used on both mobile and web browsers with no issues.

6.Julia:

Julia is an open-source, high-level, general-purpose dynamic programming language. It is both functional and object-oriented and is favored by developers as it has easy syntax. It is accessible and easily understandable.

Julia also provides support for integrated development environments (IDE) such as Visual studio and Juno and editors including Emacs and VIM. Julia's code is universally executable meaning once a code is written in a machine learning application then can be compiled in Julia intrinsically from other languages such as Python or R using PyCall or RCall. Its code is compiled at a run time or Just-in-Time through the low-level virtual machine (LLVM) framework.

Julia has the following powerful tools:

- **Flux**: It comes with the same functionality as Tensorflow. Lightweight ML library, useful tools to help you use the full power of Julia
- **Knet**: Written in Julia, active community. Deep learning framework that supports GPU operation and automatic differentiation using dynamic computational graphs for models in Julia
- **MLBase.jl**: It is used for data processing and manipulation, evaluation of models, cross-validation, and model tuning.

- **TensorFlow.jl**: It offers the option to express computations as data flow graphs.
- **ScikitLearn.jl**: It allows preprocessing, clustering, model selection.

7.Scala:

Scalable Language (Scala) is known to be a multi-paradigm programming language. It is a high-level language including features of both functional programming and object-oriented programming. Similar to Python, which is an object-oriented language, each value is an object. Some of the popular libraries of Scala that are meant for the development of applications and operations such as scientific computing, linear algebra, and random number generation are as follows:

- **Saddle**: It is built on top of array-backed data structures and is useful for data manipulation by way of 2D data structures, and array-backed support.
- **Aerosol**: Aerosol is a fast GPU and CPU-accelerated library which accelerates programming.
- **Breeze**: It is the primary scientific computing library that is very fast and efficient for manipulations with data arrays.
- **Scalalab**: Scalalab is the Sacla version of MATLAB having computing functionality. It comes with the scalability and power of Scala.
- **NLP**: NLP is used for natural language processing and has high speed and GPU usage which can parse sentences.

Tools:

1. TensorFlow:

TensorFlow is one of the most popular open-source libraries used to train and build both machine learning and deep learning models. It provides a JS library and was developed by **Google Brain Team.** It is much popular among machine learning enthusiasts, and they use it for building different ML applications. It offers a powerful library, tools, and resources for numerical computation, specifically for large scale machine learning and deep learning projects.

For training and building the ML models, TensorFlow provides a high-level Keras API, which lets users easily start with TensorFlow and machine learning.

Features:

Below are some top features:

- TensorFlow enables us to build and train our ML models easily.
- It also enables you to run the existing models using **the TensorFlow.js**
- It provides multiple abstraction levels that allow the user to select the correct resource as per the requirement.
- It helps in building a neural network.
- Provides support of distributed computing.

- While building a model, for more need of flexibility, it provides eager execution that enables immediate iteration and intuitive debugging.
- This is open-source software and highly flexible.
- It also enables the developers to perform numerical computations using data flow graphs.
- Run-on GPUs and CPUs, and also on various mobile computing platforms.
- It provides a functionality of auto diff (Automatically computing gradients is called automatic differentiation or auto diff).
- It enables to easily deploy and training the model in the cloud.
- It can be used in two ways, i.e., by installing through NPM or by script tags.
- It is free to use.

2. PyTorch:

PyTorch is an open-source machine learning framework, which is based on **the Torch** library. This framework is free and open-source and developed by **FAIR(Facebook's AI Research lab**). It is one of the popular ML frameworks, which can be used for various applications, including computer vision and natural language processing. PyTorch has Python and C++ interfaces; however, the Python interface is more interactive. Different deep learning software is made up on top of PyTorch, such as PyTorch Lightning, Hugging Face's Transformers, Tesla autopilot, etc.

It specifies a Tensor class containing an n-dimensional array that can perform tensor computations along with GPU support.

Features:

Below are some top features:

- It enables the developers to create neural networks using Autograde Module.
- It is more suitable for deep learning researches with good speed and flexibility.
- It can also be used on cloud platforms.
- It includes tutorial courses, various tools, and libraries.
- It also provides a dynamic computational graph that makes this library more popular.
- It allows changing the network behaviour randomly without any lag.
- It is easy to use due to its hybrid front-end.
- It is freely available.

3. Google Cloud ML Engine:

While training a classifier with a huge amount of data, a computer system might not perform well. However, various machine learning or deep learning projects requires millions or billions of training datasets. Or the algorithm that is being used is taking a long time for execution. In such a case, one should go for the Google Cloud ML Engine. It is a hosted platform where ML developers and data scientists build and run optimum quality machine, learning models. It provides a managed service that allows developers to easily create ML models with any type of data and of any size.

Features:

Below are the top features:

- Provides machine learning model training, building, deep learning and predictive modelling.
- The two services, namely, prediction and training, can be used independently or combinedly.
- It can be used by enterprises, i.e., for identifying clouds in a satellite image, responding faster to emails of customers.
- It can be widely used to train a complex model.

4. Amazon Machine Learning (AML):

Amazon provides a great number of machine learning tools, and one of them is **Amazon Machine Learning** or AML. Amazon Machine Learning (AML) is a cloud-based and robust machine learning software application, which is widely used for building machine learning models and making predictions. Moreover, it integrates data from multiple sources, including **Redshift, Amazon S3, or RDS.**

Features

Below are some top features:

- AML offers visualization tools and wizards.
- Enables the users to identify the patterns, build mathematical models, and make predictions.
- It provides support for three types of models, which are multi-class classification, binary classification, and regression.

- It permits users to import the model into or export the model out from Amazon Machine Learning.
- It also provides core concepts of machine learning, including ML models, Data sources, Evaluations, Real-time predictions and Batch predictions.
- It enables the user to retrieve predictions with the help of batch APIs for bulk requests or real-time APIs for individual requests.

5. NET:

Accord.Net is .Net based Machine Learning framework, which is used for scientific computing. It is combined with audio and image processing libraries that are written in C#. This framework provides different libraries for various applications in ML, such as **Pattern Recognition, linear algebra, Statistical Data processing.** One popular package of the Accord.Net framework is **Accord. Statistics, Accord.Math, and Accord.MachineLearning**.

Features

Below are some top features:

- It contains 38+ kernel Functions.
- Consists of more than 40 non-parametric and parametric estimation of statistical distributions.
- Used for creating production-grade computer audition, computer vision, signal processing, and statistics apps.
- Contains more than 35 hypothesis tests that include two-way and one way ANOVA tests, non-parametric tests such as the Kolmogorov-Smirnov test and many more.

6. Apache Mahout:

Apache Mahout is an open-source project of Apache Software Foundation, which is used for developing machine learning applications mainly focused on Linear Algebra. It is a distributed linear algebra framework and mathematically expressive Scala DSL, which enable the developers to promptly implement their own algorithms. It also provides Java/Scala libraries to perform Mathematical operations mainly based on linear algebra and statistics.

Features:

Below are some top features:

- It enables developers to implement machine learning techniques, including recommendation, clustering, and classification.
- It is an efficient framework for implementing scalable algorithms.
- It consists of matrix and vector libraries.
- It provides support for multiple distributed backends(including Apache Spark)
- It runs on top of Apache Hadoop using the MapReduce paradigm.

7. Shogun:

Shogun is a free and open-source machine learning software library, which was created by **Gunnar Raetsch and Soeren Sonnenburg** in the year **1999**. This software library is written in C++ and supports interfaces for different languages such as Python, R, Scala, C#, Ruby, etc., using **SWIG**(Simplified Wrapper and Interface Generator). The main aim of Shogun is on different kernel-based algorithms such as Support Vector Machine (SVM), K-Means Clustering, etc., for regression and classification problems. It also provides the complete implementation of Hidden Markov Models.

Features:

Below are some top features:

- The main aim of Shogun is on different kernel-based algorithms such as Support Vector Machine (SVM), K-Means Clustering, etc., for regression and classification problems.
- It provides support for the use of pre-calculated kernels.
- It also offers to use a combined kernel using Multiple kernel Learning Functionality.
- This was initially designed for processing a huge dataset that consists of up to 10 million samples.
- It also enables users to work on interfaces on different programming languages such as Lua, Python, Java, C#, Octave, Ruby, MATLAB, and R.

8. Oryx2:

It is a realization of the lambda architecture and built on **Apache Kafka** and **Apache Spark**. It is widely used for real-time large-scale machine learning projects. It is a framework for building apps, including end-to-end applications for filtering, packaged, regression, classification, and clustering. It is written in Java languages, including Apache Spark, Hadoop, Tomcat, Kafka, etc. The latest version of Oryx2 is Oryx 2.8.0.

Features:

Below are some top features:

- It has three tiers: specialization on top providing ML abstractions, generic lambda architecture tier, end-to-end implementation of the same standard ML algorithms.
- The original project of Oryx2 was Oryx1, and after some upgrades, Oryx2 was launched.
- It is well suited for large-scale real-time machine learning projects.
- It contains three layers which are arranged side-by-side, and these are named as Speed layer, batch layer, and serving layer.
- It also has a data transport layer that transfer data between different layers and receives input from external sources.

9. Apache Spark MLlib:

Apache Spark MLlib is a scalable machine learning library that runs on Apache Mesos, Hadoop, Kubernetes, standalone, or in the cloud. Moreover, it can access data from different data sources. It is an open-source cluster-computing framework that offers an interface for complete clusters along with data parallelism and fault tolerance.

For optimized numerical processing of data, MLlib provides linear algebra packages such as Breeze and netlib-Java. It uses a query optimizer and physical execution engine for achieving high performance with both batch and streaming data.

Features

Below are some top features:

- MLlib contains various algorithms, including Classification, Regression, Clustering, recommendations, association rules, etc.
- It runs different platforms such as Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud against diverse data sources.
- It contains high-quality algorithms that provide great results and performance.
- It is easy to use as it provides interfaces In Java, Python, Scala, R, and SQL.

10. Google ML kit for Mobile:

For Mobile app developers, Google brings ML Kit, which is packaged with the expertise of machine learning and technology to create more robust, optimized, and personalized apps. This tools kit can be used for face detection, text recognition, landmark detection, image labelling, and barcode scanning applications. One can also use it for working offline.

Features:

Below are some top features:

- The ML kit is optimized for mobile.
- It includes the advantages of different machine learning technologies.
- It provides easy-to-use APIs that enables powerful use cases in your mobile apps.
- It includes Vision API and Natural Language APIS to detect faces, text, and objects, and identify different languages & provide reply suggestions.

Issues:

1. Inadequate Training Data:

The major issue that comes while using machine learning algorithms is the lack of quality as well as quantity of data. Although data plays a vital role in the processing of machine learning algorithms, many data scientists claim that inadequate data, noisy data, and unclean data are extremely exhausting the machine learning algorithms. For example, a simple task requires thousands of sample data, and an advanced task such as speech or image recognition needs millions of sample data examples. Further, data quality is also important for the algorithms to work ideally, but the absence of data quality is also found in Machine Learning applications. Data quality can be affected by some factors as follows:

- **Noisy Data-** It is responsible for an inaccurate prediction that affects the decision as well as accuracy in classification tasks.
- **Incorrect data-** It is also responsible for faulty programming and results obtained in machine learning models. Hence, incorrect data may affect the accuracy of the results also.
- **Generalizing of output data-** Sometimes, it is also found that generalizing output data becomes complex, which results in comparatively poor future actions.

2. Poor quality of data:

As we have discussed above, data plays a significant role in machine learning, and it must be of good quality as well. Noisy data, incomplete data, inaccurate data, and unclean data lead to less accuracy in classification and low-quality results. Hence, data quality can also be considered as a major common problem while processing machine learning algorithms.

3. Non-representative training data:

To make sure our training model is generalized well or not, we have to ensure that sample training data must be representative of new cases that we need to generalize. The training data must cover all cases that are already occurred as well as occurring.

A machine learning model is said to be ideal if it predicts well for generalized cases and provides accurate decisions. If there is less training data, then there will be a sampling noise in the model, called the non-representative training set. It won't be accurate in predictions. To overcome this, it will be biased against one class or a group.

Hence, we should use representative data in training to protect against being biased and make accurate predictions without any drift.

4. Overfitting and Underfitting:

Overfitting:

Overfitting is one of the most common issues faced by Machine Learning engineers and data scientists. Whenever a machine learning model is trained with a huge amount of data, it starts capturing noise and inaccurate data into the training data set. It negatively affects the performance of the model. Let's understand with a simple example where we have a few training data sets such as 1000 mangoes, 1000 apples, 1000 bananas, and 5000 papayas. Then there is a considerable probability of identification of an apple as papaya because we have a massive amount of biased data in the training data set; hence prediction got negatively affected. The main reason behind overfitting is using non-linear methods used in machine learning algorithms as they build non-realistic data models. We can overcome overfitting by using linear and parametric algorithms in the machine learning models.

Methods to reduce overfitting:

- Increase training data in a dataset.
- Reduce model complexity by simplifying the model by selecting one with fewer parameters
- Ridge Regularization and Lasso Regularization
- Early stopping during the training phase
- Reduce the noise
- Reduce the number of attributes in training data.
- Constraining the model.

Underfitting:

Underfitting is just the opposite of overfitting. Whenever a machine learning model is trained with fewer amounts of data, and as a result, it provides incomplete and inaccurate data and destroys the accuracy of the machine learning model.

Underfitting occurs when our model is too simple to understand the base structure of the data, just like an undersized pant. This generally happens when we have limited data into the data set, and we try to build a linear model with non-linear data. In such scenarios, the complexity of the model destroys, and rules of the machine learning model become too easy to be applied on this data set, and the model starts doing wrong predictions as well.

Methods to reduce Underfitting:

- Increase model complexity
- Remove noise from the data
- Trained on increased and better features
- Reduce the constraints
- Increase the number of epochs to get better results.

5. Monitoring and maintenance:

As we know that generalized output data is mandatory for any machine learning model; hence, regular monitoring and maintenance become compulsory for the same. Different results for different actions require data change; hence editing of codes as well as resources for monitoring them also become necessary.

6. Getting bad recommendations:

A machine learning model operates under a specific context which results in bad recommendations and concept drift in the model. Let's understand with an example where at a specific time customer is looking for some gadgets, but now customer requirement changed over time but still machine learning model showing same recommendations to the customer while customer expectation has been changed. This incident is called a Data Drift. It generally occurs when new data is introduced or interpretation of data changes. However, we can overcome this by regularly updating and monitoring data according to the expectations.

7. Lack of skilled resources:

Although Machine Learning and Artificial Intelligence are continuously growing in the market, still these industries are fresher in comparison to others. The absence of skilled resources in the form of manpower is also an issue. Hence, we need manpower having in-depth knowledge of mathematics, science, and technologies for developing and managing scientific substances for machine learning.

8. Customer Segmentation”

Customer segmentation is also an important issue while developing a machine learning algorithm. To identify the customers who paid for the recommendations shown by the model and who don't even check them. Hence, an algorithm is necessary to recognize the customer behavior and trigger a relevant recommendation for the user based on past experience.

9. Process Complexity of Machine Learning:

The machine learning process is very complex, which is also another major issue faced by machine learning engineers and data scientists. However, Machine Learning and Artificial Intelligence are very new technologies but are still in an experimental phase and continuously being changing over time. There is the majority of hits and trial experiments; hence the probability of error is higher than expected. Further, it also includes analyzing the data, removing data bias, training data, applying complex mathematical calculations, etc., making the procedure more complicated and quite tedious.

10. Data Bias:

Data Biasing is also found a big challenge in Machine Learning. These errors exist when certain elements of the dataset are heavily weighted or need more importance than others. Biased data leads to inaccurate results, skewed outcomes, and other analytical errors. However, we can resolve this error by determining where data is actually biased in the dataset. Further, take necessary steps to reduce it.

Methods to remove Data Bias:

- Research more for customer segmentation.
- Be aware of your general use cases and potential outliers.
- Combine inputs from multiple sources to ensure data diversity.
- Include bias testing in the development process.
- Analyze data regularly and keep tracking errors to resolve them easily.
- Review the collected and annotated data.
- Use multi-pass annotation such as sentiment analysis, content moderation, and intent recognition.

Preparing to Model:

Before deploying a machine learning model, it is important to prepare the data to ensure that it is in the correct format and that any errors or inconsistencies have been cleaned. Here are some steps to prepare data before deploying a machine learning model:

1. **Data collection:** Collect the data that you will use to train your model. This could be from a variety of sources such as databases, CSV files, or APIs.
2. **Data cleaning**: Check for any missing, duplicate or inconsistent data and clean it. This may include removing any irrelevant columns, filling in missing values, and formatting data correctly.
3. **Data exploration:** Explore the data to gain insights into its distribution, relationships between features, and any outliers. Use visualization tools to help identify patterns, anomalies and trends.
4. **Data preprocessing:** Prepare the data for use in the model by normalizing or scaling the data, and transforming it into a format that the model can understand.
5. **Data splitting:** Divide the data into training, validation, and testing sets. The training set is used to train the model, the validation set is used to fine-tune the model, and the testing set is used to evaluate the model's performance.
6. **Data augmentation:** This step is optional, but it can help to improve the model's performance by creating new examples from the existing data. This can include techniques such as rotating, flipping, or cropping images.
7. **Data annotation:** This step is also optional, but it's important when working with image, video or audio data. Annotating the data is the process of labeling the data, for example, by bounding boxes, polygons, or points, to indicate the location of objects in the data.
8. It's also important to note that before deployment, the data should be checked for any bias and take action to remove bias or mitigate its effect.

Introduction of Preparing Model:

What is Data Preparation:

Data preparation is defined as a gathering, combining, cleaning, and transforming raw data to make accurate predictions in Machine learning projects.

Data preparation is also known as data "pre-processing," "data wrangling," "data cleaning," "data pre-processing," and "feature engineering." It is the later stage of the machine learning lifecycle, which comes after data collection.

Data preparation is particular to data, the objectives of the projects, and the algorithms that will be used in data modeling techniques.

Prerequisites for Data Preparation:

Everyone must explore a few essential tasks when working with data in the data preparation step. These are as follows:

- **Data cleaning:** This task includes the identification of errors and making corrections or improvements to those errors.

- **Feature Selection:** We need to identify the most important or relevant input data variables for the model.
- **Data Transforms:** Data transformation involves converting raw data into a well-suitable format for the model.
- **Feature Engineering:** Feature engineering involves deriving new variables from the available dataset.
- **Dimensionality Reduction:** The dimensionality reduction process involves converting higher dimensions into lower dimension features without changing the information.

Data Preparation in Machine Learning:

Data Preparation is the process of cleaning and transforming raw data to make predictions accurately through using ML algorithms. Although data preparation is considered the most complicated stage in ML, it reduces process complexity later in real-time projects. Various issueshave been reported during the data preparation step in machine learning as follows:

- **Missing data:** Missing data or incomplete records is a prevalent issue found in most datasets. Instead of appropriate data, sometimes records contain empty cells, values (e.g., NULL or N/A), or a specific character, such as a question mark, etc.
- **Outliers or Anomalies:** ML algorithms are sensitive to the range and distribution of values when data comes from unknown sources. These values can spoil the entire machine learning training system and the performance of the model. Hence, it is essential to detect these outliers or anomalies through techniques such as visualization technique.
- **Unstructured data format:** Data comes from various sources and needs to be extracted into a different format. Hence, before deploying an ML project, always consult with domain experts or import data from known sources.
- **Limited Features:** Whenever data comes from a single source, it contains limited features, so it is necessary to import data from various sources for feature enrichment or build multiple features in datasets.
- **Understanding feature engineering:** Features engineering helps develop additional content in the ML models, increasing model performance and accuracy in predictions.

Why is Data Preparation important?

Each machine learning project requires a specific data format. To do so, datasets need to be prepared well before applying it to the projects. Sometimes, data in data sets have missing or incomplete information, which leads to less accurate or incorrect predictions. Further, sometimes data sets are clean but not adequately shaped, such as aggregated or pivoted, and some have less business context. Hence, after collecting data from various data sources, data preparation needs to transform raw data. Below are a few significant advantages of data preparation in machine learning as follows:

- It helps to provide reliable prediction outcomes in various analytics operations.
- It helps identify data issues or errors and significantly reduces the chances of errors.
- It increases decision-making capability.
- It reduces overall project cost (data management and analytic cost).
- It helps to remove duplicate content to make it worthwhile for different applications.
- It increases model performance.

Steps in Data Preparation Process:

Data preparation is one of the critical steps in the machine learning project building process, and it must be done in particular series of steps which includes different tasks. There are some essential steps of the data preparation process in machine learning suggested by different ML experts and professionals as follows:

1. **Understand the problem:** This is one of the essential steps of data preparation for a machine learning model in which we need to understand the actual problem and try to solve it. To build a better model, we must have detailed information on all issues, such as what to do and how to do it. It is also very much effective to retain clients without wasting much effort.
2. **Data collection:** Data collection is probably the most typical step in the data preparation process, where data scientistsneed to collect data from various potential sources. These data sources may be either within enterprise or third parties vendors. Data collection is beneficial to reduce and mitigate biasing in the ML model; hence before collecting data, always analyze it and also ensure that the data set was collected from diverse people, geographical areas, and perspectives.

   There are some common problems that can be addressed using data collection as follows:

   - It is helpful to determine the relevant attributes in the string for the .csv file format.

- It is used to parse highly nested data structures files such as XML or JSON into tabular form.
- It is significant in easier scanning and pattern detection in data sets.
- Data collection is a practical step in machine learning to find relevant data from external repositories.

3. **Profiling and Data Exploration:** After analyzing and collecting data from various data sources, it's time to explore data such as trends, outliers, exceptions, incorrect, inconsistent, missing, or skewed information, etc. Although source data will provide all model findings, it does not contain unseen biases. Data exploration helps to determine problems such as collinearity, which means a situation when the Standardization of data sets and other data transformations are necessary.
4. **Data Cleaning and Validation:** Data cleaning and validation techniques help determine and solve inconsistencies, outliers, anomalies, incomplete data, etc. Clean data helps to find valuable patterns and information in data and ignoresirrelevant data in the datasets. It is very much essential to build high-quality models, and missing or incomplete data is one of the best examples of poor data. Since missing data always reduces prediction accuracy and performance of the model, data must be cleaned and validated through various imputation tools to fill incomplete fields with statistically relevant substitutes.
5. **Data Formatting:** After cleaning and validating data, the following approach is to ensure that the data is correctly formatted or not. If data is formatted incorrectly, it will help build a high-quality model.Since data comes from various sources or is sometimes updated manually, there are high chances of discrepancies in the data format. For example, if you have collected data from two sources, one source has updated the product's price to USD10.50, and the other has updated the same value to $10.50. Similarly, there may be anomalies in their spelling, abbreviation, etc. This type of data formation leads to incorrect predictions. To reduce these errors, you must format your data inconsistent manner by using some input formatting protocols.
6. **Improve data quality:** Quality is one of the essential parameters in building high-quality models. Quality data helps to reduce errors, missing data, extreme values, and outliers in the datasets. We can understand it with an example such, In one dataset, columns have First Name and Last NAME, and another dataset has Column named as a customer that combines First and Last Name. Then in such cases, intelligent ML algorithms must have

the ability to match these columns and join the dataset for a singular view of the customer.

7. **Feature engineering and selection:** Feature engineering is defined as the study of selecting, manipulating, and transforming raw data into valuable features or most relevant variables in supervised machine learning.Feature engineering enables you to build an enhanced predictive model with accurate predictions.

   For example, data can be spitted into various parts to capture more specific information, such as analyzingmarketing performance by the day of the week, not only the month or year. In this situation, segregating the day as a separate categorical value from the data (e.g., "Mon; 07.12.2021") may provide the algorithm with more relevant information.

   There are various feature engineering techniques used in machine learning as follows:

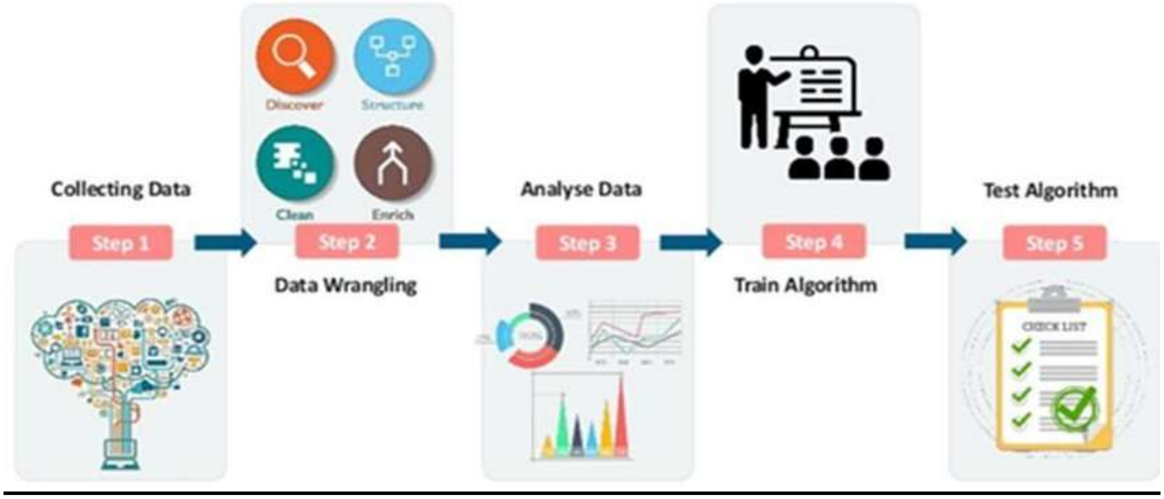
   - **Imputation:** Feature imputation is the technique to fill incomplete fields in the datasets. It is essential because most machine learning models don't work when there are missing data in the dataset. Although, the missing values problem can be reduced by using techniques such as single value imputation, multiple value imputation, K-Nearest neighbor, deleting the row, etc.
   - **Encoding:** Feature encoding is defined as the method to convert string values into numeric form. This is important as all ML models require all values in numeric format. Feature encoding includes label encoding and One Hot Encoding (also known as get_dummies).

   Similarly, feature engineering also includes handling outliers, log transform, scaling, normalization, Standardization, etc.

8. **Splitting data:** After feature engineering and selection, the last step is to split your data into two different sets (training and evaluation sets). Further, always select non-overlapping subsets of your data for the training and evaluation sets to ensure proper testing.

Machine Learning Activities:

The task of imparting intelligence to machines seems daunting and impossible. But it is actually really easy. It can be broken down into 7 major steps :



1. Collecting Data:

As you know, machines initially learn from the data that you give them. It is of the utmost importance to collect reliable data so that your machine learning model can find the correct patterns. The quality of the data that you feed to the machine will determine how accurate your model is. If you have incorrect or outdated data, you will have wrong outcomes or predictions which are not relevant.

Make sure you use data from a reliable source, as it will directly affect the outcome of your model. Good data is relevant, contains very few missing and repeated values, and has a good representation of the various subcategories/classes present.

2. Preparing the Data:

After you have your data, you have to prepare it. You can do this by :

- Putting together all the data you have and randomizing it. This helps make sure that data is evenly distributed, and the ordering does not affect the learning process.
- Cleaning the data to remove unwanted data, missing values, rows, and columns, duplicate values, data type conversion, etc. You might even have to restructure the dataset and change the rows and columns or index of rows and columns.
- Visualize the data to understand how it is structured and understand the relationship between various variables and classes present.
- Splitting the cleaned data into two sets - a training set and a testing set. The training set is the set your model learns from. A testing set is used to check the accuracy of your model after training.

3. Choosing a Model:

A machine learning model determines the output you get after running a machine learning algorithm on the collected data. It is important to choose a model which is relevant to the task at hand. Over the years, scientists and engineers developed various models suited for different tasks like speech recognition, image recognition, prediction, etc. Apart from this, you also have to see if your model is suited for numerical or categorical data and choose accordingly.

4. Training the Model:

Training is the most important step in machine learning. In training, you pass the prepared data to your machine learning model to find patterns and make predictions. It results in the model learning from the data so that it can accomplish the task set. Over time, with training, the model gets better at predicting.

5. Evaluating the Model:

After training your model, you have to check to see how it's performing. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that you split our data into earlier. If testing was done on the same data which is used for training, you will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give you disproportionately high accuracy.

When used on testing data, you get an accurate measure of how your model will perform and its speed.

6.Parameter Tuning:

Once you have created and evaluated your model, see if its accuracy can be improved in any way. This is done by tuning the parameters present in your model. Parameters are the variables in the model that the programmer generally decides. At a particular value of your parameter, the accuracy will be the maximum. Parameter tuning refers to finding these values

7. Making Predictions:

In the end, you can use your model on unseen data to make predictions accurately.

Types of data:

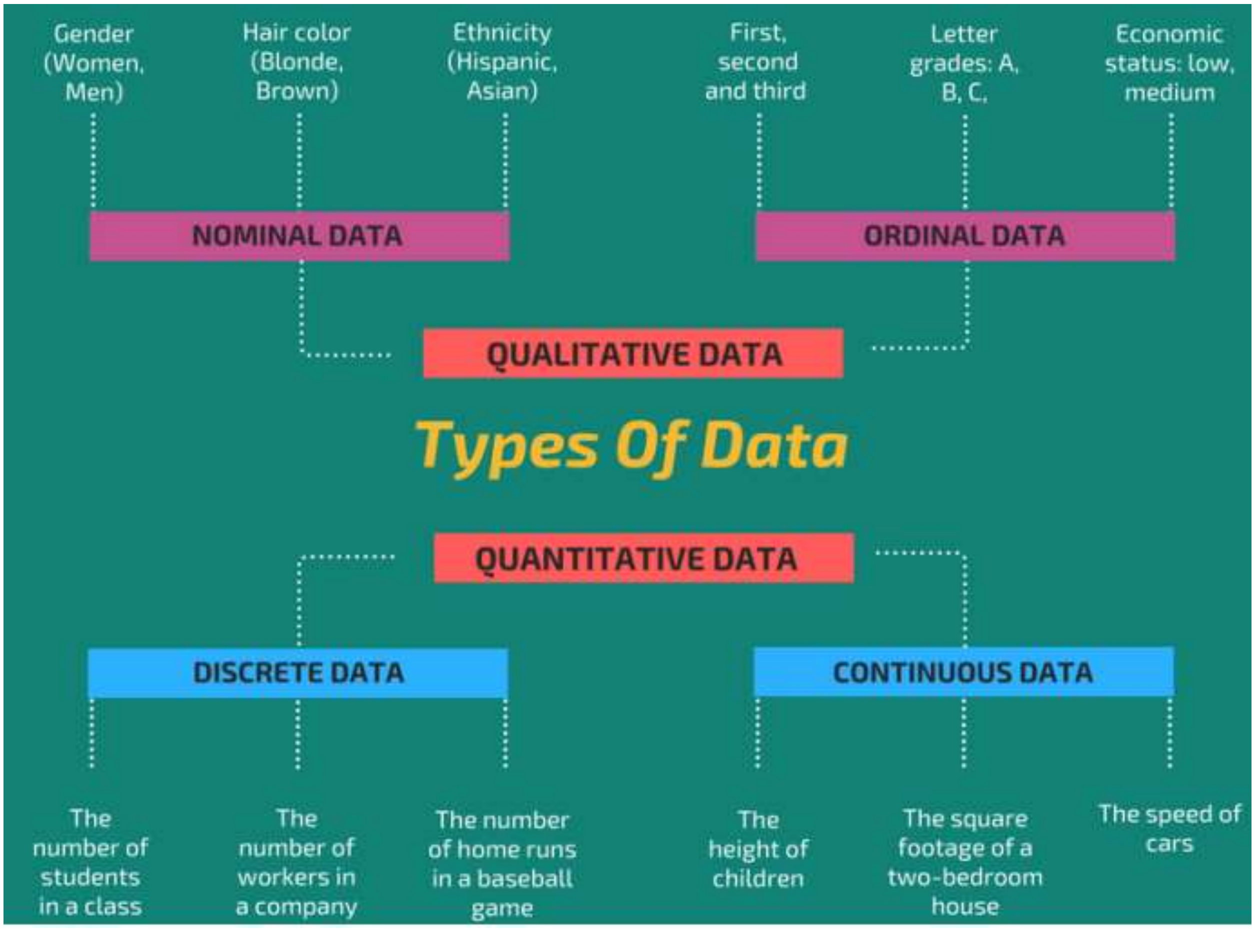
Data Types Are A Way Of Classification That Specifies Which Type Of Value A Variable Can Store And What Type Of Mathematical Operations, Relational, Or Logical Operations Can Be Applied To The Variable Without Causing An Error. In Machine Learning, It Is Very Important To Know Appropriate Datatypes Of Independent And Dependent Variable.

As It Provides The Basis For Selecting Classification Or Regression Models. Incorrect Identification Of Data Types Leads To Incorrect Modeling Which In Turn Leads To An Incorrect Solution.

Here I Will Be Discussing Different Types Of Data Types With Suitable Examples.

Different Types Of Data Types
The Data Type Is Broadly Classified Into

1. Quantitative or **Numerical**
2. Qualitative or **Categorical**.


Gender (Women, Men)
Hair color (Blonde, Brown)
Ethnicity (Hispanic, Asian)
First, second and third
Letter grades: A, B, C,
Economic status: low, medium
NOMINAL DATA
ORDINAL DATA
QUALITATIVE DATA
Types Of Data
QUANTITATIVE DATA
DISCRETE DATA
CONTINUOUS DATA
The number of students in a class
The number of workers in a company
The number of home runs in a baseball game
The height of children
The square footage of a two-bedroom house
The speed of cars

Quantitative Data:

Quantitative data can be expressed in numerical values, making it countable and including statistical data analysis. These kinds of data are also known as Numerical data. It answers the questions like “how much,” “how many,” and “how often.” For example, the price of a phone, the computer’s ram, the height or weight of a person, etc., falls under quantitative data.

Quantitative data can be used for statistical manipulation. These data can be represented on a wide variety of graphs and charts, such as bar graphs, histograms, scatter plots, boxplots, pie charts, line graphs, etc.

Examples of Quantitative Data :

- Height or weight of a person or object
- Room Temperature
- Scores and Marks (Ex: 59, 80, 60, etc.)
- Time

The Quantitative data are further classified into two parts :

Discrete Data:

The term discrete means distinct or separate. The discrete data contain the values that fall under integers or whole numbers. The total number of students in a class is an example of discrete data. These data can't be broken into decimal or fraction values.

The discrete data are countable and have finite values; their subdivision is not possible. These data are represented mainly by a bar graph, number line, or frequency table.

Examples of Discrete Data :

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

Continuous Data:

Continuous data are in the form of fractional numbers. It can be the version of an android phone, the height of a person, the length of an object, etc. Continuous data represents information that can be divided into smaller levels. The continuous variable can take any value within a range.

The key difference between discrete and continuous data is that discrete data contains the integer or whole number. Still, continuous data stores the fractional numbers to record different types of data such as temperature, height, width, time, speed, etc.

Examples of Continuous Data :

- Height of a person

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

Difference between Discrete and Continuous Data:

Discrete Data	**Continuous Data**
Discrete data are countable and finite; they are whole numbers or integers	Continuous data are measurable; they are in the form of fractions or decimal
Discrete data are represented mainly by bar graphs	Continuous data are represented in the form of a histogram
The values cannot be divided into subdivisions into smaller pieces	The values can be divided into subdivisions into smaller pieces
Discrete data have spaces between the values	Continuous data are in the form of a continuous sequence
Examples: Total students in a class, number of days in a week, size of a shoe, etc	**Example:** Temperature of room, the weight of a person, length of an object, etc

Qualitative or Categorical Data:

Qualitative or Categorical Data is data that can’t be measured or counted in the form of numbers. These types of data are sorted by category, not by number. That’s why it is also known as Categorical Data. These data consist of audio, images, symbols, or text. The gender of a person, i.e., male, female, or others, is qualitative data.

Qualitative data tells about the perception of people. This data helps market researchers understand the customers’ tastes and then design their ideas and strategies accordingly.

The other examples of qualitative data are :

- What language do you speak
- Favorite holiday destination
- Opinion on something (agree, disagree, or neutral)
- Colors

The Qualitative data are further classified into two parts :

Nominal Data:

Nominal Data is used to label variables without any order or quantitative value. The color of hair can be considered nominal data, as one color can't be compared with another color.

The name "nominal" comes from the Latin name "nomen," which means "name." With the help of nominal data, we can't do any numerical tasks or can't give any order to sort the data. These data don't have any meaningful order; their values are distributed into distinct categories.

Examples of Nominal Data :

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

Ordinal Data:

Ordinal data have natural ordering where a number is present in some kind of order by their position on the scale. These data are used for observation like customer satisfaction, happiness, etc., but we can't do any arithmetical tasks on them.

Ordinal data is qualitative data for which their values have some kind of relative position. These kinds of data can be considered "in-between" qualitative and quantitative data. The ordinal data only shows the sequences and cannot use for statistical analysis. Compared to nominal data, ordinal data have some kind of order that is not present in nominal data.

Examples of Ordinal Data :

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

- Education Level (Higher, Secondary, Primary)

Difference between Nominal and Ordinal Data:

Nominal Data	Ordinal Data
Nominal data can’t be quantified, neither they have any intrinsic ordering	Ordinal data gives some kind of sequential order by their position on the scale
Nominal data is qualitative data or categorical data	Ordinal data is said to be “in-between” qualitative data and quantitative data
They don’t provide any quantitative value, neither can we perform any arithmetical operation	They provide sequence and can assign numbers to ordinal data but cannot perform the arithmetical operation
Nominal data cannot be used to compare with one another	Ordinal data can help to compare one item with another by ranking or ordering
Examples: Eye color, housing style, gender, hair color, religion, marital status, ethnicity, etc	**Examples:** Economic status, customer satisfaction, education level, letter grades, etc

Exploring structure of data:

What is Data Structure?

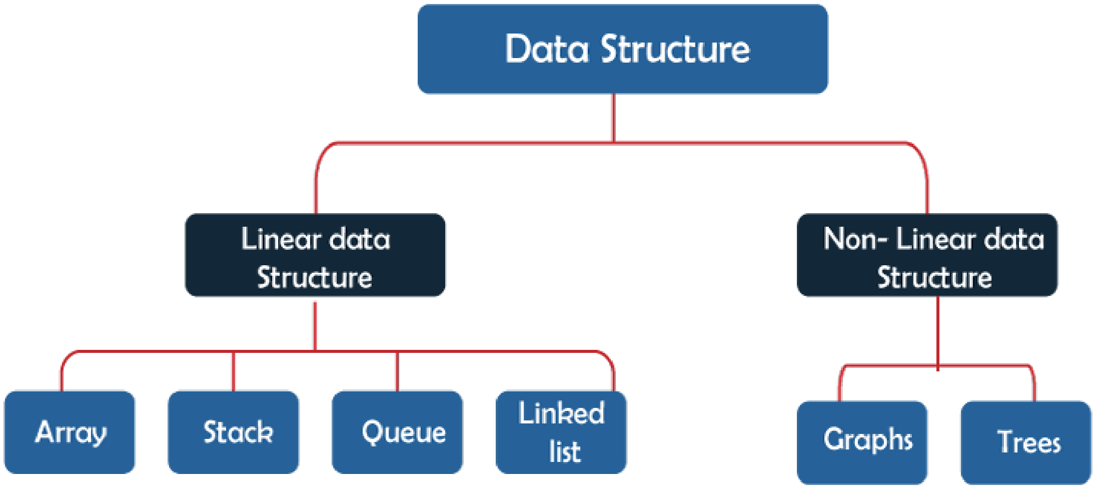
The data structure is defined as the basic building block of computer programming that helps us to organize, manage and store data for efficient search and retrieval.

In other words, the data structure is the collection of data type 'values' which are stored and organized in such a way that it allows for efficient access and modification.

Types of Data Structure:

The data structure is the ordered sequence of data, and it tells the compiler how a programmer is using the data such as **Integer, String, Boolean, etc.**

There are two different types of data structures: Linear and Non-linear data structures.



Now let's discuss popular data structures used for Machine Learning:

1. Linear Data structure:

The linear data structure is a special type of data structure that helps to organize and manage data in a specific order where the elements are attached adjacently.

There are mainly 4 types of linear data structure as follows:

Array:

An array is one of the most basic and common data structures used in Machine Learning. It is also used in linear algebra to solve complex mathematical problems. You will use arrays constantly in machine learning, whether it's:

- To convert the column of a data frame into a list format in pre-processing analysis
- To order the frequency of words present in datasets.
- Using a list of tokenized words to begin clustering topics.
- In word embedding, by creating multi-dimensional matrices.

An array contains index numbers to represent an element starting from 0. The lowest index is arr[0] and corresponds to the first element.

Stacks:

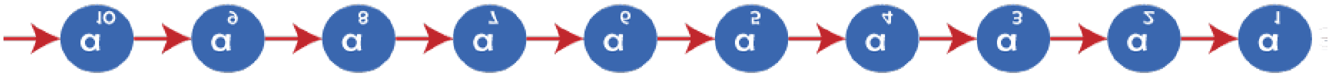
Stacks are based on the concept of LIFO (Last in First out) or FILO (First In Last Out). ***It is* used for binary classification in deep learning.** *Although stacks are easy to learn and implement* in ML models but having a good grasp can help in many computer science aspects such as parsing grammar, etc.

Stacks enable the **undo** and **redo** buttons on your computer as they function similar to a stack of blog content. There is no sense in adding a blog at the bottom of the stack. However, we can only check the most recent one that has been added. Addition and removal occur at the top of the stack.

Linked List:

A linked list is the type of collection having several separately allocated nodes. Or in other words, **a list is the type of collection of data elements that consist of a value and pointer that point to the next node in the list.**

In a linked list, insertion and deletion are constant time operations and are very efficient, but accessing a value is slow and often requires scanning. So, a linked list is very significant for a dynamic array where the shifting of elements is required. Although insertion of an element can be done at the head, middle or tail position, it is relatively cost consuming. However, linked lists are easy to splice together and split apart. Also, the list can be converted to a fixed-length array for fast access.



Queue:

A Queue is defined as the "FIFO" (first in, first out). It is useful to predict a queuing scenario in real-time programs, such as people waiting in line to withdraw cash in the bank. Hence, the queue is significant in a program where multiple lists of codes need to be processed.

The queue data structure can be used to record the split time of a car in F1 racing.

2. Non-linear Data Structures:

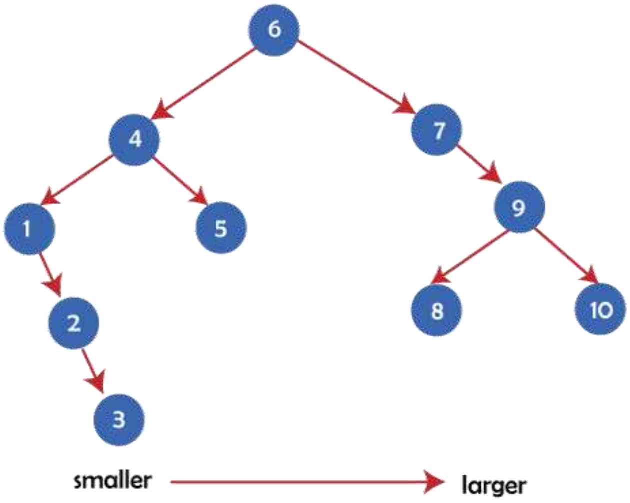
As the name suggests, in Non-linear data structures, elements are not arranged in any sequence. All the elements are arranged and linked with each other in a hierarchal manner, where one element can be linked with one or more elements.

1) Trees:

Binary Tree:

The concept of a binary tree is very much similar to a linked list, but the only difference of nodes and their pointers. In a linked list, each node contains a data value with a pointer that points to the next node in the list, whereas; ***in a binary tree, each node has two pointers to subsequent nodes instead of just one***.

Binary trees are sorted, so insertion and deletion operations can be easily done with O(log N) time complexity. Similar to the linked list, a binary tree can also be converted to an array on the basis of tree sorting.



In a binary tree, there are some child and parent nodes shown in the above image. Where the value of the left child node is always less than the value of the parent node while the value of the right-side child nodes is always more than the parent node. Hence, in a binary tree structure, data sorting is done automatically, which makes insertion and deletion efficient.

2) Graphs:

A graph data structure is also very much useful in machine learning for link prediction. Graphs are directed or undirected concepts with nodes and ordered or unordered pairs. Hence, you must have good exposure to the graph data structure for machine learning and deep learning.

3) Maps:

Maps are the popular data structure in the programming world, which are mostly useful for minimizing the run-time algorithms and fast searching the data. It stores data in the form of (key, value) pair, where the key must be unique; however, the value can be duplicated. Each key corresponds to or maps a value; hence it is named a Map.

In different programming languages, core libraries have built-in maps or, rather, HashMaps with different names for each implementation.

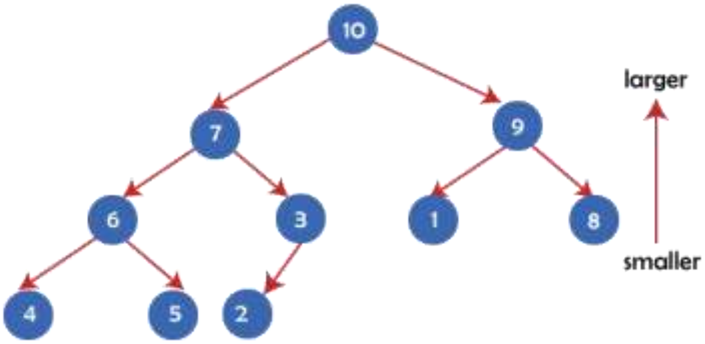
- **In Java: Maps**
- **In Python: Dictionaries**
- **C++: hash_map, unordered_map, etc.**

Python Dictionaries are very useful in machine learning and data science as various functions and algorithms return the dictionary as an output. Dictionaries are also much used for implementing sparse matrices, which is very common in Machine Learning.

4) Heap data structure:

Heap is a hierarchically ordered data structure. Heap data structure is also very much similar to a tree, but it consists of vertical ordering instead of horizontal ordering.

Ordering in a heap DS is applied along the hierarchy but not across it, where the value of the parent node is always more than that of child nodes either on the left or right side.



Here, the insertion and deletion operations are performed on the basis of promotion. It means, firstly, the element is inserted at the highest available position. After that, it gets compared with its parent and promoted until it reaches the correct ranking position. Most of the heaps data structures can be stored in an array along with the relationships between the elements.

Dynamic array data structure:

This is one of the most important types of data structure used in linear algebra to solve 1-D, 2-D, 3-D as well as 4-D arrays for matrix arithmetic. Further, it requires good exposure to Python libraries such as **Python NumPy** for programming in deep learning.

How is Data Structure used in Machine Learning?

For a Machine learning professional, apart from knowledge of machine learning skills, it is required to have mastery of data structure and algorithms.

When we use machine learning for solving a problem, we need to evaluate the model performance, i.e., which model is fastest and requires the smallest amount of space and resources with accuracy. Moreover, if a model is built using algorithms, comparing and contrasting two algorithms to determine the best for the job is crucial to the machine learning professional. For such cases, skills in data structures become important for ML professionals.

Data quality and remediation:

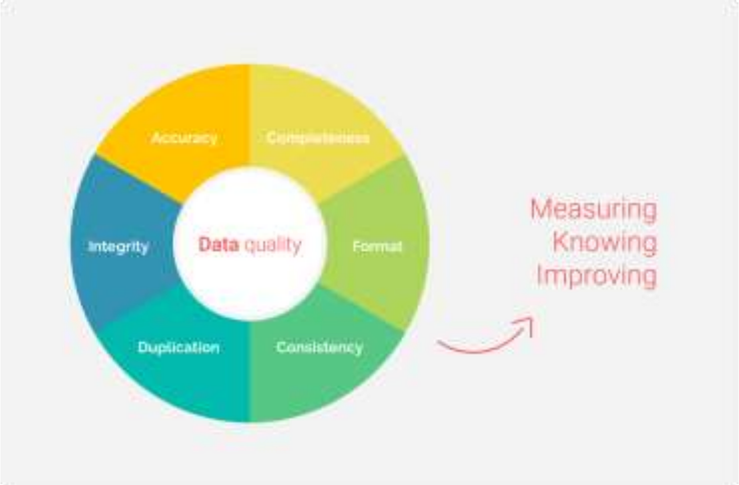
Data Quality:

Data quality is the measure of how well suited a data set is to serve its specific purpose. Measures of data quality are based on data quality characteristics such as accuracy, completeness, consistency, validity, uniqueness, and timeliness.

Data quality refers to the development and implementation of activities that apply quality management techniques to data in order to ensure the data is fit to serve the specific needs of an organization in a particular context. Data that is deemed fit for its intended purpose is considered high quality data.

Examples of data quality issues include duplicated data, incomplete data, inconsistent data, incorrect data, poorly defined data, poorly organized data, and poor data security.

Data quality rules are an integral component of data governance, which is the process of developing and establishing a defined, agreed-upon set of rules and standards by which all data across an organization is governed.



Data Quality Dimensions:

By which metrics do we measure data quality? There are six main dimensions of data quality: accuracy, completeness, consistency, validity, uniqueness, and timeliness.

Accuracy: The data should reflect actual, real-world scenarios; the measure of accuracy can be confirmed with a verifiable source.

Completeness: Completeness is a measure of the data’s ability to effectively deliver all the required values that are available.

Consistency: Data consistency refers to the uniformity of data as it moves across networks and applications. The same data values stored in difference locations should not conflict with one another.

Validity: Data should be collected according to defined business rules and parameters, and should conform to the right format and fall within the right range.

Uniqueness: Uniqueness ensures there are no duplications or overlapping of values across all data sets. Data cleansing and deduplication can help remedy a low uniqueness score.

Timeliness: Timely data is data that is available when it is required. Data may be updated in real time to ensure that it is readily available and accessible.

How to Improve Data Quality:

Data quality measures can be accomplished with data quality tools, which typically provide data quality management capabilities such as:

Data profiling - The first step in the data quality improvement process is understanding your data. Data profiling is the initial assessment of the current state of the data sets.

Data Standardization - Disparate data sets are conformed to a common data format.

Geocoding - The description of a location is transformed into coordinates that conform to U.S. and worldwide geographic standards

Matching or Linking - Data matching identifies and merges matching pieces of information in big data sets.

Data Quality Monitoring - Frequent data quality checks are essential. Data quality software in combination with machine learning can automatically detect, report, and correct data variations based on predefined business rules and parameters.

Batch and Real time - Once the data is initially cleansed, an effective data quality framework should be able to deploy the same rules and processes across all applications and data types at scale.

Data Quality vs Data Integrity:

Data quality oversight is just one component of data integrity. Data integrity refers to the process of making data useful to the organization. The four main components of data integrity include:

Data Integration: data from disparate sources must be seamlessly integrated.

Data Quality: Data must be complete, unique, valid, timely, consistent, and accurate.

Location Intelligence: Location insights adds a layer of richness to data and makes it more actionable.

Data Enrichment: Data enrichment adds a more complete, contextualized view of data by adding data from external sources, such as customer data, business data, location data, etc

Data Quality Assurance vs Data Quality Control:

Data quality assurance is the process of identifying and eliminating anomalies by means of data profiling and cleansing. Data quality control is performed both before and after quality assurance, and entails the means by which data usage for an application is controlled.

The quality control process is important for detecting duplicates, outliers, errors, and missing information. Some real-life data quality examples include:
Healthcare: accurate, complete, and unique patient data is essential for facilitating risk management and fast and accurate billing.

Public Sector: accurate, complete, and consistent data is essential to track the progress of current projects and proposed initiatives.

Financial Services: Sensitive financial data must be identified and protected, reporting processes must be automated, and regulatory compliances must be remediated.

Manufacturing: Accurate customer and vendor data must be maintained in order to track spending, reduce operational costs, and create alerts for quality assurance issues and maintenance needs.

Why Data Quality is Important to an Organization:

An increasing number of organizations are using data to inform their decisions regarding marketing, product development, communications strategies and more. High quality data can be processed and analyzed quickly, leading to better and faster insights that drive business

intelligence efforts and big data analytics.
Good data quality management helps extract greater value from data sets, and contributes to reduced risks and costs, increased efficiency and productivity, more informed decision-making, better audience targeting, more effective marketing campaigns, better customer relations, and an overall stronger competitive edge.

Data remediation:

What is data remediation:

Data remediation is the process of cleansing, organizing and migrating data so that it's properly protected and best serves its intended purpose. There is a misconception that data remediation simply means deleting business data that is no longer needed. It's important to remember that the key word "remediation" derives from the word "remedy," which is to correct a mistake. Since the core initiative is to correct data, the data remediation process typically involves replacing, modifying, cleansing or deleting any "dirty" data.

Data remediation terminology:

As you explore the data remediation process, you will come across unique terminology. These are common terms related to data remediation that you should get acquainted with.

- **Data Migration** – The process of moving data between two or more systems, data formats or servers.
- **Data Discovery** – A manual or automated process of searching for patterns in data sets to identify structured and unstructured data in an organization's systems.
- **ROT** – An acronym that stands for redundant, obsolete and trivial data. According to the Association for Intelligent Information Management, ROT data accounts for nearly 80 percent of the unstructured data that is beyond its recommended retention period and no longer useful to an organization.
- **Dark Data** – Any information that businesses collect, process and store, but do not use for other purposes. Some examples include customer call records, raw survey data or email correspondences. Often, the storing and securing of this type of data incurs more expense and sometimes even greater risk than it does value.
- **Dirty Data** – Data that damages the integrity of the organization's complete dataset. This can include data that is unnecessarily duplicated, outdated, incomplete or inaccurate.
- **Data Overload** – This is when an organization has acquired too much data, including low-quality or dark data. Data overload makes the tasks of identifying, classifying and remediating data laborious.
- **Data Cleansing** – Transforming data in its native state to a predefined standardized format.

- **Data Governance** – Management of the availability, usability, integrity and security of the data stored within an organization.

Stages of data remediation:

Data remediation is an involved process. After all, it's more than simply purging your organization's systems of dirty data. It requires knowledgeable assessment on how to most effectively resolve unclean data.

Assessment:

Before you take any action on your company's data, you need to have a complete understanding of the data you possess. How valuable is this data to the company? Is this data sensitive? Does this data actually require specialized storage, or is it trivial information? Identifying the quantity and type of data you're dealing with, even if it's just a ballpark estimate to start, will help your team get a general sense of how much time and resources need to be dedicated for successful data remediation.

Organizing and segmentation:

Not all data is created equally, which means that not all pieces of data require the same level of protection or storage features. For instance, it isn't cost-efficient for a company to store all data, ranging from information that is publicly facing to sensitive data, all in the same high-security vault. This is why organizing and creating segments based on the information's purpose is critical during the data remediation process.

Accessibility is a big factor to consider when it comes to segmenting data. There's data that needs to be easily accessed by team members for day-to-day tasks, and then there's data that needs to have higher security measures for legal or regulatory purposes. This is one example of two segments an organization may create.

ROT data is a good example of information that can be safely deleted, while other business records that are still within a recommended retention period could be stored in an archive system.

Indexation and classification:

Once your data is segmented, you can move onto indexing and classification. These steps build off of the data segments you have created and helps you determine action steps. In this step,organizations will focus on segments containing non-ROT data and classify the level of sensitivity of this remaining data.

“Restricted data” is a common sensitive data classification term for data of this nature. Then, there’s unregulated and unstructured data that may consider sensitive information, and could be classified as internal, confidential or restricted data, depending on its level of sensitivity.

Migrating:

If an organization’s end goal is to consolidate their data into a new, cleansed storage environment, then migration is an essential step in the data remediation process. A common scenario is an organization who needs to find a new secure location for storing data because their legacy system has reached its end of life. Some organizations may also prefer moving their data to cloud-based platforms, like SharePoint or Office 365, so that information is more accessible for their internal teams.

Data cleansing:

The final task for your organization’s data may not always involve migration. There may be other actions better suited for the data depending on what segmentation group it falls under and its classification. A few vital actions that a team may proceed with include shredding, redacting, quarantining, ACL removal and script execution to clean up data.

Business benefits of data remediation:

Data remediation is a big effort, but it comes with big benefits for businesses as well. These are the top benefits that most organizations realize after data remediation.

- **Reduced data storage costs** — Although data remediation isn’t solely about deletion of data, it is a common remediation action and less data means less storage required. Additionally, many organizations realize that they have lumped trivial information in the same high-security storage platform for sensitive information, instead of only paying for the storage space that’s actually necessary.
- **Protection for unstructured sensitive data** — Once sensitive data is discovered and classified, remediation is where you determine and execute the actions that mitigate risk. This could look like finding a secure area to store sensitive data or deleting what is necessary from a compliance perspective.
- **Reduced sensitive data footprint** — By removing sensitive data that is beyond its recommended retention period and is necessary for compliance, you’ve reduced your organization’s sensitive data footprint and decreased risk of potential data breaches or leaks of highly sensitive data.

- **Adherence to compliance laws and regulations** — Hanging on to data that is beyond its recommended retention period can create greater risks. By cleaning up data, your organization reduces data exposure which supports compliance initiatives.
- **Increased staff productivity** — Data that your team uses should be available, usable and trustworthy. By streamlining your organization's network with data remediation, information should be easier to find and usable for its intended purpose.
- **Minimized cyberattack risks** — By continuously engaging in data remediation, your organization is proactively minimizing data loss risks and potential financial or reputational damage of successful cyberattacks.
- **Improved overall data security** — Data remediation and data governance work hand in hand. In order to properly remediate data, your organization will need to establish data governance policies, which is significant for the overall management and protection of your organization's data.

When is data remediation necessary?

Data remediation is an essential process for any organization to ensure optimal hygiene and legal compliance standing. It's recommended for any company to stay consistent with data remediation, but there are some specific instances that may occur and become a strong driver for prioritizing data remediation.

Business changes:

If a company has changed software or systems they use, or even moved to a new office or data center location, that is a case to buckle down on data remediation immediately. Sometimes companies switch to new softwares or systems because they need to phase out their legacy system that has reached its end of life. Change of any kind is rarely ever 100 percent smooth, and data could become corrupted or exposed during the shuffle of changing environments — whether it be digital or physical.

Even if your organization's data is pristine, you cannot say the same about the new company that is joining forces with you until you take the time to discover, classify and, eventually, remediate data.

Laws and regulations:

Newly enacted laws or regulations, either on a state or federal level, could be another major driver for data remediation. Data privacy and protection laws are continuously being updated and improved upon, like the more recent California Consumer Protection Act of 2018 (CCPA). Sometimes new policies may be enacted by the leadership team at your organization as well.

Human error:

Drivers for data remediation aren't always necessarily as grand as a new business acquisition or legal regulation. Sometimes, instances as simple as human error can be a catalyst for data remediation. For instance, let's say that your organization discovers one of its employees has unintentionally downloaded sensitive corporate data on their personal mobile phone. Or, perhaps a couple of employees accidentally opened up a malicious spam email. Actions as innocent as these examples could put the integrity of your organization's data at risk and is cause for immediately taking action with data remediation.

More examples of scenarios that may trigger the need to remediate data include:

- Preparing legal documentation for an investor portfolio sale
- Eliminating personally identifiable information (PII) or personal healthcare information (PHI)
- Enterprise resource planning (ERP)
- Master Data Management (MDM) implementation

What prevents organizations from performing data remediation?

As important as data remediation is, many organizations bypass this process. Oftentimes, other activities like data migration may seem to be an adequate replacement for the exhaustive task of comprehensive data remediation. However, projects like that are typically one-time endeavors that aren't a continuous effort of cleansing and validating an organization's data.

Lack of information:

A common reason that organizations ignore data remediation is a lack of information about what, where, how and why data is stored in the company. An organization may not even realize the expanse of data they have collected or where it's even stored. It's recommended that organizations, especially those who belong to industries that interact with high volumes of sensitive data (like the medical, financial or education industries), regularly perform sensitive data discovery and data classification to prepare for data remediation. All of these steps are essential to a healthy data lifecycle and depend on one another to keep a company's data security in good standing.

Fear of deleting data:

Another factor that may prevent an organization from getting started with data remediation is a fear of deleting data. The permanency of the action can be intimidating, and some businesses may be concerned that they may need the data at hand at some point in the future. However, hanging on to unnecessary data, or leaving dirty data unmodified or uncleansed, can pose greater risk to an organization — especially when it comes to compliance laws and regulations.

Unclear data ownership:

Lastly, some organizations may not have established clear data ownership. If there aren't clear roles and responsibilities for each member of your organization's security team, then important tasks like data remediation can easily slip through the cracks. It's essential to determine each person's key responsibilities when it comes to maintaining data security, and to make those duties transparent across the organization so that everyone knows who to turn to for specific security questions, and to keep the team accountable.

How to prepare your business for data remediation:

Whether you've put data remediation on the back-burner or are realizing for the first time the benefits of steady data remediation, here are several steps your team should take to prepare for data remediation.

1. **Data remediation teams** – First, create data remediation teams. In doing this, your organization will need to establish data ownership roles and responsibilities, so everyone on your security team knows how they are contributing and who to go for with questions or concerns.
2. **Data governance policies** – From there, you will need to establish company policies that enforce data governance. An effective data governance plan will ensure that the company's data is trustworthy and does not get misused. Typically, data governance is a process largely based on the company's internal data standards and policies that control data usage in order to maintain the availability, usability, integrity and security of data.
3. **Prioritize data remediation areas** – Once you have your organization's policies and data remediation team assembled, you should begin prioritizing which areas may require more immediate data remediation. If any of the drivers we mentioned above have occurred, such as your organization switching to a new platform or an urgent need to

eliminate PII, those are great starting points for prioritizing the order of business areas that need data remediation.

4. **Budget for data-related issues** – After compiling a prioritized list, it's time to budget for any data-related issues that may occur during the remediation process. This includes estimating the hours of labor for the process and factoring in costs for any special tools that may be needed for remediation.
5. **Discuss data remediation expectations** – Either after or alongside the budgeting process, your team should sit down and discuss general expectations of the data remediation process. Are there any types of sensitive data your team expects to find? Are there any recent overarching data security issues or changes that could have an impact or effect on the remediation process? During the discussion, important details may be brought to light for the team that only one person was aware of and help the team reach success.
6. **Track progress and ROI** – All company's want to understand their ROI on big projects and initiatives, and this applies to data security measures too. Your organization's IT data security lead should create a progress reporting mechanism that can inform company stakeholders on the data remediation progress, including key performance indicators like amount of issues resolved or how resolved issues translate into money and risk saved.

Data Pre-processing:

What Is Data Preprocessing:

Data preprocessing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers and machine learning.

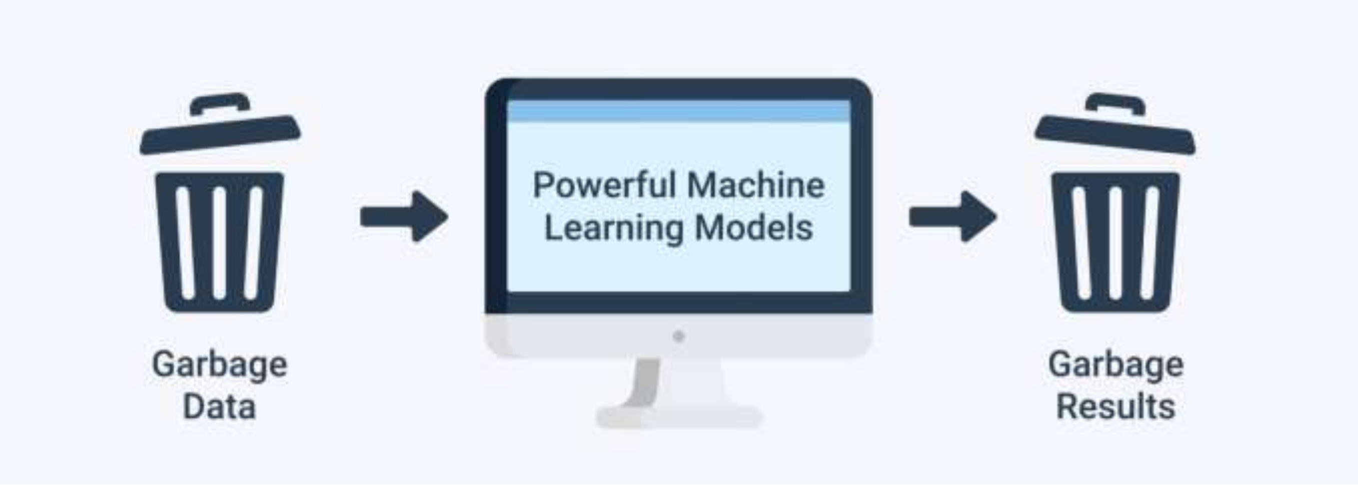
Raw, real-world data in the form of text, images, video, etc., is messy. Not only may it contain errors and inconsistencies, but it is often incomplete, and doesn't have a regular, uniform design.

Machines like to process nice and tidy information – they read data as 1s and 0s. So calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis.

Data Preprocessing Importance:

When using data sets to train machine learning models, you'll often hear the phrase **"garbage in, garbage out"** This means that if you use bad or "dirty" data to train your model, you'll end up with a bad, improperly trained model that won't actually be relevant to your analysis.

Good, preprocessed data is even more important than the most powerful algorithms, to the point that machine learning models trained with bad data could actually be harmful to the analysis you're trying to do – giving you "garbage" results.



Depending on your data gathering techniques and sources, you may end up with data that's out of range or includes an incorrect feature, like household income below zero or an image from a set of "zoo animals" that is actually a tree. Your set could have missing values or fields. Or text data, for example, will often have misspelled words and irrelevant symbols, URLs, etc.

When you properly preprocess and clean your data, you'll set yourself up for much more accurate downstream processes. We often hear about the importance of "data-driven decision making," but if these decisions are driven by bad data, they're simply bad decisions.

Understanding Machine Learning Data Features:

Data sets can be explained with or communicated as the "features" that make them up. This can be by size, location, age, time, color, etc. Features appear as columns in datasets and are also known as attributes, variables, fields, and characteristics.

Wikipedia describes a machine learning data feature as **"an individual measurable property or characteristic of a phenomenon being observed"**.

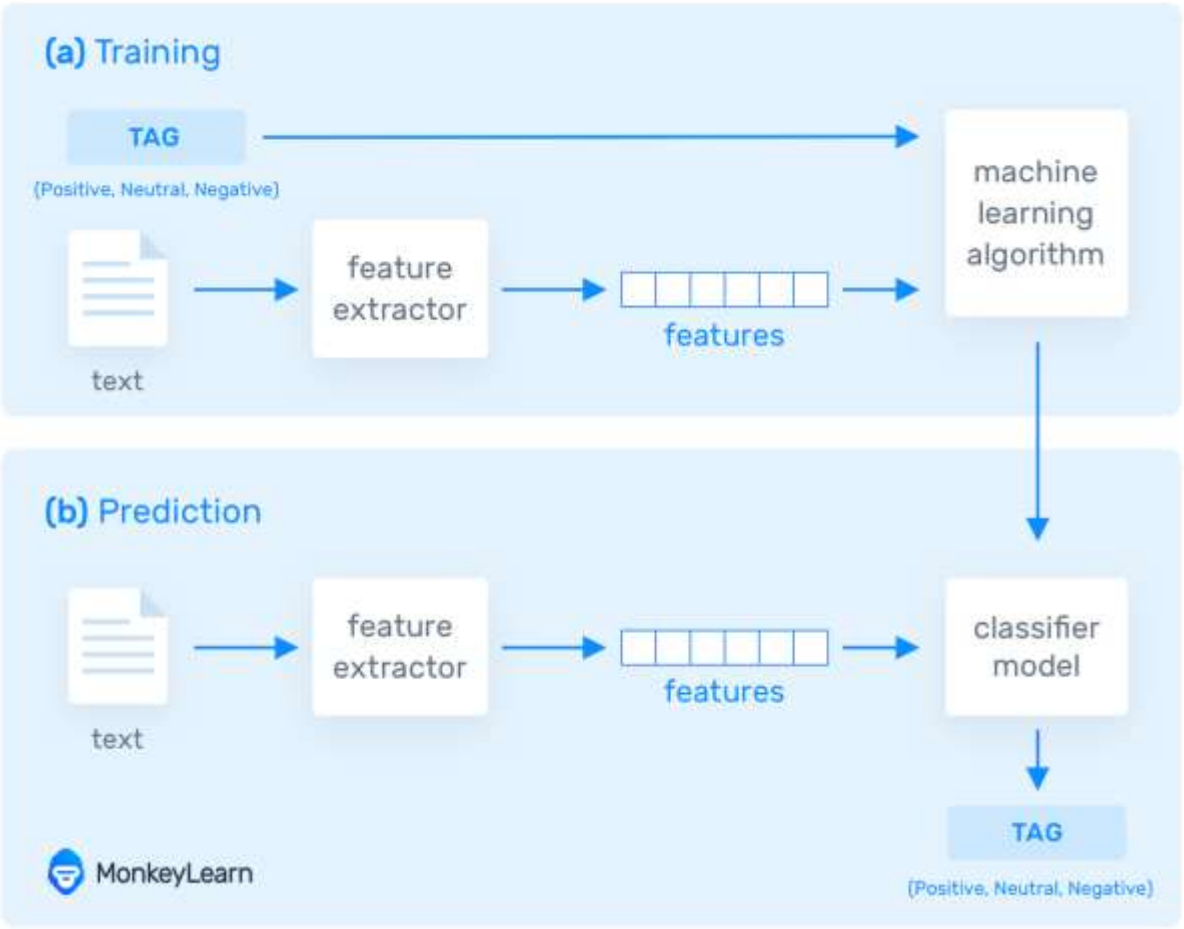
It's important to understand what "features" are when preprocessing your data because you'll need to choose which ones to focus on depending on what your business goals are. Later, we'll explain how you can improve the quality of your dataset's features and the insights you gain with processes like feature selection

First, let's go over the two different types of features that are used to describe data: categorical and numerical:

- **Categorical features:** Features whose explanations or values are taken from a defined set of possible explanations or values. Categorical values can be colors of a house; types of animals; months of the year; True/False; positive, negative, neutral, etc. The set of possible categories that the features can fit into is predetermined.

- **Numerical features:** Features with values that are continuous on a scale, statistical, or integer-related. Numerical values are represented by whole numbers, fractions, or percentages. Numerical features can be house prices, word counts in a document, time it takes to travel somewhere, etc.

The diagram below shows how features are used to train machine learning text analysis models. Text is run through a feature extractor (to pull out or highlight words or phrases) and these pieces of text are classified or tagged by their features. Once the model is properly trained, text can be run through it, and it will make predictions on the features of the text or "tag" the text itself.



Data Preprocessing Steps:

Let's take a look at the established steps you'll need to go through to make sure your data is successfully preprocessed.

1. **Data quality assessment**

2. **Data cleaning**
3. **Data transformation**
4. **Data reduction**

1. Data quality assessment:

Take a good look at your data and get an idea of its overall quality, relevance to your project, and consistency. There are a number of data anomalies and inherent problems to look out for in almost any data set, for example:

- **Mismatched data types:** When you collect data from many different sources, it may come to you in different formats. While the ultimate goal of this entire process is to reformat your data for machines, you still need to begin with similarly formatted data. For example, if part of your analysis involves family income from multiple countries, you'll have to convert each income amount into a single currency.
- **Mixed data values:** Perhaps different sources use different descriptors for features – for example, *man* or *male*. These value descriptors should all be made uniform.
- **Data outliers:** Outliers can have a huge impact on data analysis results. For example if you're averaging test scores for a class, and one student didn't respond to any of the questions, their 0% could greatly skew the results.
- **Missing data:** Take a look for missing data fields, blank spaces in text, or unanswered survey questions. This could be due to human error or incomplete data. To take care of missing data, you'll have to perform data cleaning.

2. Data cleaning:

Data cleaning is the process of adding missing data and correcting, repairing, or removing incorrect or irrelevant data from a data set. Dating cleaning is the most important step of preprocessing because it will ensure that your data is ready to go for your downstream needs.

Data cleaning will correct all of the inconsistent data you uncovered in your data quality assessment. Depending on the kind of data you're working with, there are a number of possible cleaners you'll need to run your data through.

Missing data:

There are a number of ways to correct for missing data, but the two most common are:

- **Ignore the tuples:** A tuple is an ordered list or sequence of numbers or entities. If multiple values are missing within tuples, you may simply discard the tuples with that missing information. This is only recommended for large data sets, when a few ignored tuples won't harm further analysis.
- **Manually fill in missing data:** This can be tedious, but is definitely necessary when working with smaller data sets.

Noisy data:

Data cleaning also includes fixing “noisy” data. This is data that includes unnecessary data points, irrelevant data, and data that’s more difficult to group together.

- **Binning:** Binning sorts data of a wide data set into smaller groups of more similar data. It’s often used when analyzing demographics. Income, for example, could be grouped: $35,000-$50,000, $50,000-$75,000, etc.
- **Regression:** Regression is used to decide which variables will actually apply to your analysis. Regression analysis is used to smooth large amounts of data. This will help you get a handle on your data, so you’re not overburdened with unnecessary data.
- **Clustering:** Clustering algorithms are used to properly group data, so that it can be analyzed with like data. They’re generally used in unsupervised learning, when not a lot is known about the relationships within your data.

If you’re working with text data, for example, some things you should consider when cleaning your data are:

- Remove URLs, symbols, emojis, etc., that aren’t relevant to your analysis
- Translate all text into the language you’ll be working in
- Remove HTML tags
- Remove boilerplate email text
- Remove unnecessary blank text between words
- Remove duplicate data

After data cleaning, you may realize you have insufficient data for the task at hand. At this point you can also perform data wrangling or data enrichment to add new data sets and run them through quality assessment and cleaning again before adding them to your original data.

3. Data transformation:

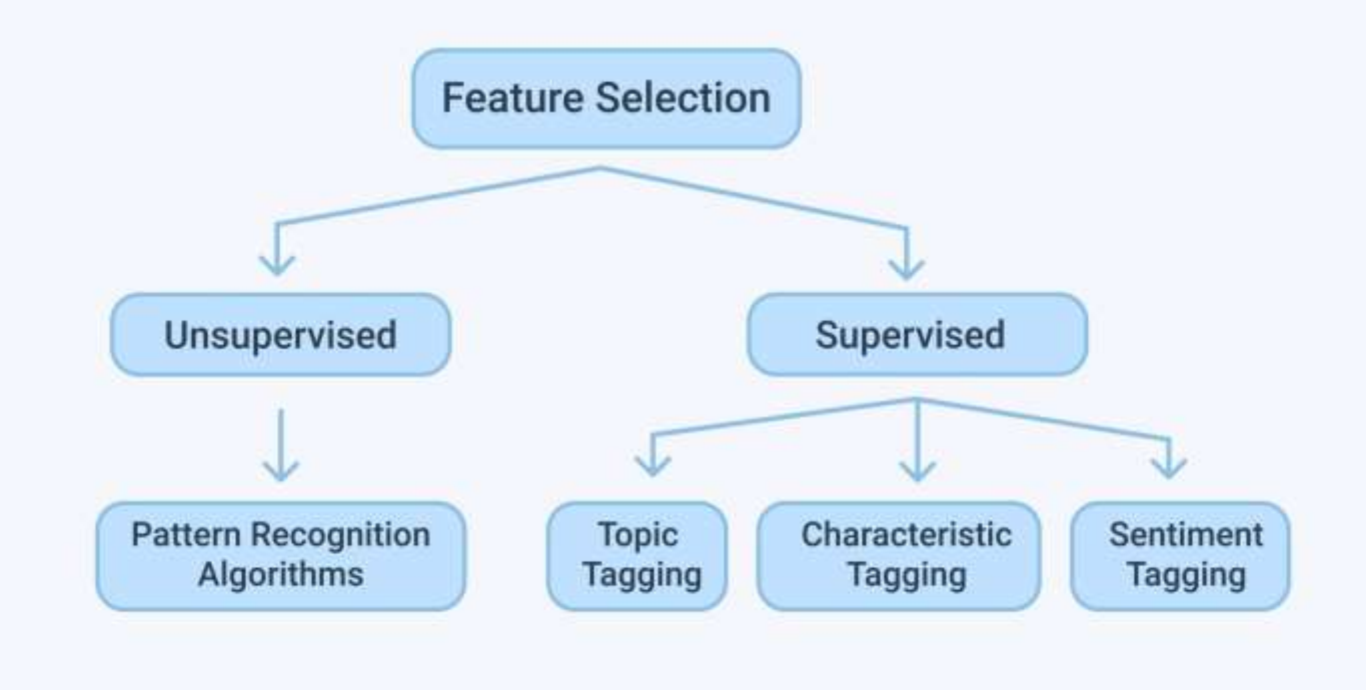
With data cleaning, we’ve already begun to modify our data, but data transformation will begin the process of turning the data into the proper format(s) you’ll need for analysis and other downstream processes.

This generally happens in one or more of the below:

1. Aggregation
2. Normalization
3. Feature selection
4. Discreditization
5. Concept hierarchy generation

- **Aggregation:** Data aggregation combines all of your data together in a uniform format.

- **Normalization:** Normalization scales your data into a regularized range so that you can compare it more accurately. For example, if you're comparing employee loss or gain within a number of companies (some with just a dozen employees and some with 200+), you'll have to scale them within a specified range, like -1.0 to 1.0 or 0.0 to 1.0.
- **Feature selection:** Feature selection is the process of deciding which variables (features, characteristics, categories, etc.) are most important to your analysis. These features will be used to train ML models. It's important to remember, that the more features you choose to use, the longer the training process and, sometimes, the less accurate your results, because some feature characteristics may overlap or be less present in the data.



- **Discreditization:** Discreditiization pools data into smaller intervals. It's somewhat similar to binning, but usually happens after data has been cleaned. For example, when calculating average daily exercise, rather than using the exact minutes and seconds, you could join together data to fall into 0-15 minutes, 15-30, etc.
- **Concept hierarchy generation:** Concept hierarchy generation can add a hierarchy within and between your features that wasn't present in the original data. If your analysis contains wolves and coyotes, for example, you could add the hierarchy for their genus: canis.

4. Data reduction:

The more data you're working with, the harder it will be to analyze, even after cleaning and transforming it. Depending on your task at hand, you may actually have more data than you need. Especially when working with text analysis, much of regular human speech is superfluous or irrelevant to the needs of the researcher. Data reduction not only makes the analysis easier and more accurate, but cuts down on data storage.

It will also help identify the most important features to the process at hand.

- **Attribute selection:** Similar to discreditization, attribute selection can fit your data into smaller pools. It, essentially, combines tags or features, so that tags like *male/female* and *professor* could be combined into *male professor/female professor*.
- **Numerosity reduction:** This will help with data storage and transmission. You can use a regression model, for example, to use only the data and variables that are relevant to your analysis.
- **Dimensionality reduction:** This, again, reduces the amount of data used to help facilitate analysis and downstream processes. Algorithms like *K-nearest neighbors* use pattern recognition to combine similar data and make it more manageable.

Data Preprocessing Examples:

Take a look at the table below to see how preprocessing works. In this example, we have three variables: name, age, and company. In the first example we can tell that #2 and #3 have been assigned the incorrect companies.

Name	Age	Company
Karen Lynch	57	CVS Health
Elon Musk	49	Amazon
Jeff Bezos	57	Tesla
Tim Cook	60	Apple

We can use data cleaning to simply remove these rows, as we know the data was improperly entered or is otherwise corrupted.

Name	Age	Company
Karen Lynch	57	CVS Health
Tim Cook	60	Apple

Or, we can perform data transformation, in this case, manually, in order to fix the problem:

Name	Age	Company
Karen Lynch	57	CVS Health
Elon Musk	49	Tesla
Jeff Bezos	57	Amazon
Tim Cook	60	Apple

Once the issue is fixed, we can perform data reduction, in this case by descending age, to choose which age range we want to focus on:

Name	Age	Company
Tim Cook	60	Apple
Karen Lynch	57	CVS Health
Jeff Bezos	57	Amazon
Elon Musk	49	Tesla