AIML

Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

## Features of Machine Learning:

* Machine learning uses data to detect various patterns in a given dataset.
* It can learn from past data and improve automatically.
* It is a data-driven technology.
* Machine learning is much similar to data mining as it also deals with the huge amount of the data

## Need for Machine Learning

* Rapid increment in the production of data
* Solving complex problems, which are difficult for a human
* Decision making in various sector including finance
* Finding hidden patterns and extracting useful information from data.

## Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

**Supervised learning**

**Unsupervised learning**

**Reinforcement learning**

### Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision,

Supervised learning can be grouped further in two categories of algorithms:

* Classification
* Regression

### Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classifieds into two categories of algorithms:

* Clustering
* Association

### Reinforcement Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

# Applications of Machine learning

### 1. Image Recognition:

### 2. Speech Recognition

### 3. Traffic prediction:

### 4. Product recommendations:

### 5. Self-driving cars:

### 6. Email Spam and Malware Filtering: 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: various virtual personal assistants such as ****Google assistant****, ****Alexa****, ****Cortana****, ****Siri****.

### 8. Online Fraud Detection:

### Key Differences Between Supervised vs Unsupervised Learning vs Reinforcement Learning

1. Supervised Learning deals with two main tasks Regression and Classification. Unsupervised Learning deals with clustering and associative rule mining problems. Whereas [Reinforcement Learning](https://www.geeksforgeeks.org/what-is-reinforcement-learning/) deals with exploitation or exploration, Markov’s decision processes, Policy Learning, [Deep Learning](https://www.aitude.com/category/deep-learning/) and value learning.
2. Supervised Learning works with the labelled data and here the output data patterns are known to the system. But, the unsupervised learning deals with unlabeled data where the output is based on the collection of perceptions. Whereas in Reinforcement Learning Markov’s Decision process- the agent interacts with the environment in discrete steps.
3. The name itself says, Supervised Learning is highly supervised. And Unsupervised Learning is not supervised. As against, Reinforcement Learning is less supervised which depends on the agent in determining the output.
4. The input data in Supervised Learning in labelled data. Whereas, in Unsupervised Learning the data is unlabelled. The data is not predefined in Reinforcement Learning.
5. Supervised Learning predicts based on a class type. Unsupervised Learning discovers underlying patterns. And in Reinforcement Learning, the learning agent works as a reward and action system.
6. Supervised learning maps labelled data to known output. Whereas, Unsupervised Learning explore patterns and predict the output. Reinforcement Learning follows a trial and error method.
7. To sum up, in Supervised Learning, the goal is to generate formula based on input and output values. In Unsupervised Learning, we find an association between input values and group them. In Reinforcement Learning an agent learn through delayed feedback by interacting with the environment.

### Comparison Table

| **Criteria** | **Supervised ML** | **Unsupervised ML** | **Reinforcement ML** |
| --- | --- | --- | --- |
| Definition | Learns by using labelled data | Trained using unlabelled data without any guidance. | Works on interacting with the environment |
| Type of data | Labelled data | Unlabelled data | No – predefined data |
| Type of problems | Regression and classification | Association and Clustering | Exploitation or Exploration |
| Supervision | Extra supervision | No supervision | No supervision |
| Algorithms | Linear Regression, Logistic Regression, SVM, KNN etc. | K – Means, C – Means, Apriori | Q – Learning, SARSA |
| Aim | Calculate outcomes | Discover underlying patterns | Learn a series of action |
| Application | Risk Evaluation, Forecast Sales | Recommendation System, Anomaly Detection | Self Driving Cars, Gaming, Healthcare |

# Machine learning Life cycle

Machine learning life cycle involves seven major steps, which are given below:

* **Gathering Data**
* **Data preparation**
* **Data Wrangling**
* **Analyse Data**
* **Train the model**
* **Test the model**
* **Deployment**

## 1. Gathering Data:

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.In this step, we need to identify the different data sources, as data can be collected from various sources such as **files**, **database**, **internet**, or **mobile devices**. It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction.

This step includes the below tasks:

* **Identify various data sources**
* **Collect data**
* **Integrate the data obtained from different sources**

By performing the above task, we get a coherent set of data, also called as a **dataset**. It will be used in further steps.

## 2. Data preparation

After collecting the data, we need to prepare it for further steps. Data preparation is a step where we put our data into a suitable place and prepare it to use in our machine learning training.In this step, first, we put all data together, and then randomize the ordering of data.

This step can be further divided into two processes:

* **Dataexploration:**  
  It is used to understand the nature of data that we have to work with. We need to understand the characteristics, format, and quality of data.  
  A better understanding of data leads to an effective outcome. In this, we find Correlations, general trends, and outliers.
* **Datapre-processing:**  
  Now the next step is preprocessing of data for its analysis.

## 3. Data Wrangling

Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete In real-world applications, collected data may have various issues, including:

* **Missing Values**
* **Duplicate data**
* **Invalid data**
* **Noise**

So, we use various filtering techniques to clean the data.

It is mandatory to detect and remove the above issues because it can negatively affect the quality of the outcome.

## 4. Data Analysis

Now the cleaned and prepared data is passed on to the analysis step. This step involves:

* **Selection of analytical techniques**
* **Building models**
* **Review the result**

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. It starts with the determination of the type of the problems, where we select the machine learning techniques such as **Classification**, **Regression**, **Cluster analysis**, **Association**, etc. then build the model using prepared data, and evaluate the model.

## 5. Train Model

Now the next step is to train the model, in this step we train our model to improve its performance for better outcome of the problem.We use datasets to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules, and, features.

## 6. Test Model

Once our machine learning model has been trained on a given dataset, then we test the model. In this step, we check for the accuracy of our model by providing a test dataset to it.Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

## 7. Deployment

The last step of machine learning life cycle is deployment, where we deploy the model in the real-world system.If the above-prepared model is producing an accurate result as per our requirement with acceptable speed, then we deploy the model in the real system. But before deploying the project, we will check whether it is improving its performance using available data or not. The deployment phase is similar to making the final report for a project.

|  | Online machine learning | Offline machine learning |
| --- | --- | --- |
| Complexity | More complex because the model keeps evolving over time as more data becomes available. | Less complex because the model is fed with more consistent data sets periodically. |
| Computational power | More computational power is required because of the continuous feed of data that leads to continuous refinement. | Fewer computational power is needed because data is delivered in batches; the model isn’t continuously refining itself. |
| Use in production | Harder to implement and control because the production model changes in real-time according to its data feed. | Easier to implement because offline learning provides engineers with more time to perfect the model before deployment. |
| Applications | Used in applications where new data patterns are constantly required (e.g., weather prediction tools) | Used in applications where data patterns remain constant and don’t have sudden concept drifts (e.g., image classification) |

‍

Question-Write a short note on instance-based learning.

Answer:

* Instance-based learning is a family of learning algorithms that, instead of performing explicit generalization, compares new problem instances with instances seen in training, which have been stored in memory.
* They are sometimes referred to as lazy learning methods because they delay processing until a new instance must be classified. The nearest neighbors of an instance are defined in terms of Euclidean distance.
* No model is learned
* The stored training instances themselves represent the knowledge
* Training instances are searched for instance that most closely resembles new instance

Instance-based learning  -(also known as **memory-based learning** or **lazy learning**) involves memorizing training data in order to make predictions about future data points. This approach doesn’t require any prior knowledge or assumptions about the data, which makes it easy to implement and understand. However, it can be computationally expensive since all of the training data needs to be stored in memory before making a prediction. Additionally, this approach doesn’t generalize well to unseen data sets because its predictions are based on memorized examples rather than learned models.

In instance-based learning, the system learns the training data by heart. At the time of making prediction, the system uses **similarity measure** and compare the new cases with the learned data. [K-nearest neighbors (KNN)](https://vitalflux.com/k-nearest-neighbors-explained-with-python-examples/) is an algorithm that belongs to the instance-based learning class of algorithms. KNN is a non-parametric algorithm because it does not assume any specific form or underlying structure in the data. Instead, it relies on a measure of similarity between each pair of data points. Generally speaking, this measure is based on either Euclidean distance or cosine similarity; however, other forms of metric can be used depending on the type of data being analyzed. Other instance-based learning algorithms include **learning vector quantization (LVQ)** and **self-organizing maps (SOMs)**. These algorithms also memorize the training examples and use them to make predictions on new data, but they use different techniques to do so.

Ques  Explain instance-based learning representation.

Answer:

Instance-based learning: It generates classification predictions using only specific instances. Instance-based learning algorithms do not maintain a set of abstractions derived from specific instances. This approach extends the nearest neighbor algorithm, which has large storage requirements.

Ques  What are the performance dimensions used for instance-based learning algorithm?

Answer:

Time complexity of Instance based learning algorithms depends upon the size of training data. Time complexity of this algorithm in the worst case is O (n), where n is the number of training items to be used to classify a single new instance.

Ques  What are the functions of instance-based learning ?

Answer:

Instance-based learning refers to a family of techniques for classification and regression, which produce a class label/prediction based on the similarity of the query to its nearest neighbor(s) in the training set.

Functions are as follows:

1. Similarity:Similarity is a machine learning method that uses a nearest neighbor approach to identify the similarity of two or more objects to each other based on algorithmic distance functions.
2. Classification:Process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories.
3. Concept Description:Much of human learning involves acquiring general concepts from past experiences.This description can then be used to predict the class labels of unlabeled cases.

Ques   What are the advantages and disadvantages of instance-based learning ?

Answer:

Advantages of instance-based learning:

* It has the ability to adapt to previously unseen data, which means that one can store a new instance or drop the old instance.

Disadvantages of instance-based learning:

* Classification costs are high.
* Large amount of memory required to store the data, and each query involves starting the identification of a local model from scratch.

Model-based learning (also known as structure-based or eager learning) takes a different approach by constructing models from the training data that can generalize better than instance-based methods. This involves using algorithms like linear regression, logistic regression, random forest, etc. trees to create an underlying model from which predictions can be made for new data points.

## Differences between Instance-based & Model-based Learning

Instance-based learning and model-based learning are two broad categories of machine learning algorithms. There are several key differences between these two types of algorithms, including:

1. **Generalization**: In **model-based learning**, the goal is to **learn a generalizable model** that can be used to make predictions on new data. This means that the model is trained on a dataset and then tested on a separate, unseen dataset to evaluate its performance. In contrast,**instance-based learning algorithms simply memorize the training examples** and use them to make predictions on new data. This means that instance-based learning algorithms don’t try to learn a generalizable model, and their performance on new data is not as reliable as model-based algorithms.
2. **Scalability**: Because**instance-based learning**algorithms simply memorize the training examples, they can be very **slow and memory-intensive** when working with large datasets. This is because the model has to store all of the training examples in memory and compare new data points to each of the stored examples. In contrast, **model-based learning algorithms can be more scalable** because they don’t have to store all of the training examples. Instead, they learn a model that can be used to make predictions without storing the training data.
3. **Interpretability**: Model-based learning algorithms often produce models that are easier to interpret than instance-based learning algorithms. This is because the model-based algorithms learn a set of rules or parameters that can be inspected to understand how the model is making predictions. In contrast, instance-based learning algorithms simply store the training examples and use them as a basis for making predictions, which can make it difficult to understand how the predictions are being made.

## What is Artificial Intelligence?

[Artificial intelligence](https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/what-is-artificial-intelligence), commonly referred to as AI, is the process of imparting data, information, and human intelligence to machines. The main goal of Artificial Intelligence is to develop self-reliant machines that can think and act like humans. These machines can mimic human behavior and perform tasks by learning and problem-solving. Most of the AI systems simulate natural intelligence to solve complex problems.

Let’s have a look at an example of an AI-driven product - Amazon Echo.

**Amazon Echo is a smart speaker that uses Alexa, the virtual assistant AI technology developed by Amazon. Amazon Alexa is capable of voice interaction, playing music, setting alarms, playing audiobooks, and giving real-time information such as news, weather, sports, and traffic reports.**

## Types of Artificial Intelligence

Reactive Machines - These are systems that only react. These systems don’t form memories, and they don’t use any past experiences for making new decisions.

Limited Memory - These systems reference the past, and information is added over a period of time. The referenced information is short-lived.

Theory of Mind - This covers systems that are able to understand human emotions and how they affect decision making. They are trained to adjust their behavior accordingly.

Self-awareness - These systems are designed and created to be aware of themselves. They understand their own internal states, predict other people’s feelings, and act appropriately.

## Applications of Artificial Intelligence

* Machine Translation such as Google Translate
* [Self Driving Vehicles](https://www.simplilearn.com/autonomous-driving-key-technologies-article) such as Google’s Waymo
* [AI Robots](https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/humanoid-robots) such as Sophia and Aibo
* Speech Recognition applications like Apple’s Siri or OK Google

## What is Deep Learning?

Deep learning is a subset of machine learning that deals with algorithms inspired by the structure and function of the human brain. [Deep learning algorithms](https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm) can work with an enormous amount of both structured and unstructured data. Deep learning’s core concept lies in artificial neural networks, which enable machines to make decisions.

The major difference between [deep learning vs machine learning](https://www.simplilearn.com/machine-learning-vs-deep-learning-major-differences-you-need-to-know-article) is the way data is presented to the machine. Machine learning algorithms usually require structured data, whereas deep learning networks work on multiple layers of [artificial neural networks.](https://www.simplilearn.com/tutorials/deep-learning-tutorial/neural-network)

## Types of Deep Neural Networks

Convolutional Neural Network (CNN) - CNN is a class of deep neural networks most commonly used for image analysis.

Recurrent Neural Network (RNN) - RNN uses sequential information to build a model. It often works better for models that have to memorize past data.

Generative Adversarial Network (GAN) - GAN are algorithmic architectures that use two neural networks to create new, synthetic instances of data that pass for real data. A GAN trained on photographs can generate new photographs that look at least superficially authentic to human observers.

Deep Belief Network (DBN) - DBN is a generative graphical model that is composed of multiple layers of latent variables called hidden units. Each layer is interconnected, but the units are not.

## How Does Deep Learning Work?

1. Calculate the weighted sums.
2. The calculated sum of weights is passed as input to the activation function.
3. The activation function takes the “weighted sum of input” as the input to the function, adds a bias, and decides whether the neuron should be fired or not.
4. The output layer gives the predicted output.
5. The model output is compared with the actual output. After training the neural network, the model uses the backpropagation method to improve the performance of the network. The cost function helps to reduce the error rate.

## Deep Learning Applications

* Cancer tumor detection
* Captionbot for captioning an image
* Music generation
* Image coloring
* Object detection

**Limitation of Machine Learning**

The benefits of machine learning translate to innovative applications that can improve the way processes and tasks are accomplished. However, despite its numerous advantages, there are still risks and challenges. Take note of the following cons or limitations of machine learning:

**1. Error diagnosis and correction:** One notable limitation of machine learning is its susceptibility to errors. Brynjolfsson and McAfee said that the actual problem with this inevitable fact is that when they do make errors, diagnosing and correcting them can be difficult because it will require going through the underlying complexities of the algorithms and associated processes.

**2. Time constraints in learning:** It is impossible to make immediate accurate predictions with a machine learning system. Remember that it learns through historical data. The bigger the data and the longer it is exposed to these data, the better it will perform. For example, using a system to play games and beat human opponents would require feeding the system with historical data and continuously exposing it to newly acquired data to make better predictions or decisions.

**3. Problems with verification:**Another limitation of machine learning is the lack of variability. It is said that machine learning deals with statistical truths rather than literal truths. In situations that are not included in the historical data, it will be difficult to prove with complete certainty that the predictions made by a machine learning system is suitable in all scenarios.

**4. Limitations of predictions:** Brynjolfsson and McAfee reminded that unlike humans, computers are not good storytellers. Machine learning systems cannot always provide rational reasons for a particular prediction or decision. They are also limited to answering questions rather than posing them. In addition, these systems does not understand context. Depending on the provided data used for training, machine learning is also prone to hidden and unintentional biases. Human input is still important to better evaluate the outputs of these systems.

**5.Time and Resources**

ML desires ample time to let the algorithms study and increase sufficient to fulfill their motive with a good sized quantity of accuracy and relevancy. It additionally wants big sources to function. This can imply extra necessities of laptop energy for you.

**6. Interpretation of Results**

Another essential assignment is the capacity to precisely interpret effects generated by means of the algorithms. You ought to additionally cautiously pick out the algorithms for your purpose.

**7. High error-susceptibility**

Machine Learning is independent however particularly inclined to errors. Suppose you instruct an algorithm with information units small adequate to now not be inclusive. You stop up with biased predictions coming from a biased coaching set. This leads to inappropriate classified ads being displayed to customers. In the case of ML, such errors can set off a chain of mistakes that can go undetected for lengthy intervals of time. And when they do get noticed, it takes pretty some time to understand the supply of the issue, and even longer to right it.

### Advantages of Machine Learning

.

#### 1. Automation of Everything

Machine Learning is responsible for cutting the workload and time. By automating things we let the algorithm do the hard work for us. Automation is now being done almost everywhere. The reason is that it is very **reliable**. Also, it helps us to think more **creatively**.

. ML is slowly transforming the industry with its automation.

#### 2. Wide Range of Applications

ML has a wide variety of applications. This means that we can apply ML on any of the major fields. ML has its role everywhere from medical, business, banking to science and tech. This helps to create more opportunities. It plays a major role in **customer interactions**.

Machine Learning can help in the detection of diseases more quickly. It is helping to lift up businesses. That is why investing in ML technology is worth it.

#### 3. Scope of Improvement

Machine Learning is the type of technology that keeps on evolving. There is a lot of scope in ML to become the top technology in the future. The reason is, it has a lot of research areas in it. This helps us to improve both **hardware** and **software**.

In hardware, we have various laptops and GPUs. These have various ML and Deep Learning networks in them. These help in the faster processing power of the system. When it comes to software we have various UIs and libraries in use. These help in designing more efficient algorithms.

#### 4. Efficient Handling of Data

Machine Learning has many factors that make it reliable. One of them is data handling. ML plays the biggest role when it comes to data at this time. It can handle any type of data.Machine Learning can be **multidimensional** or different types of data. It can process and analyze these data that normal systems can’t. Data is the most important part of any Machine Learning model. Also, studying and handling of data is a field in itself.

#### 5. Best for Education and Online Shopping

ML would be the best tool for education in the future. It provides very creative techniques to help students study. In online shopping, the ML model studies your searches. Based on your search history, it would provide advertisements. These will be about your **search** preferences in previous searches. In this, the search history is the data for the model. This is a great way to improve e-commerce with ML.

# Data Preprocessing in Machine learning

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

## Why do we need Data Preprocessing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

* **Getting the dataset**
* **Importing libraries**
* **Importing datasets**
* **Finding Missing Data**
* **Encoding Categorical Data**
* **Splitting dataset into training and test set**
* **Feature scaling**

## Which are the Data Preprocessing Techniques?

The data preprocessing techniques in machine learning can be broadly segmented into two parts: Data Cleaning and Data Transformation

**Data Cleaning/ Cleansing**

* Making the data consistent across the values, which can mean:
  + The attributes may have incorrect data types and are not in sync with the data dictionary. Correction of the data types is a must before proceeding with any type of data cleaning.
  + Replace the special characters for example: replace $ and comma signs in the column of Sales/Income/Profit i.e making $10,000 as 10000.
  + Making the format of the date column consistent with the format of the tool used for data analysis.
* Check for null or missing values, also check for the negative values. The relevancy of the negative values depends on the data. In the income column, a negative value is spurious though the same negative value in the profit column becomes a loss.
* Smoothing of the noise present in the data by identifying and treating for outliers.
* Errors may also occur at the

#### Handling the Null/Missing Values

The null values in the dataset are imputed using mean/median or mode based on the type of data that is missing:

* **Numerical Data:** If a numerical value is missing, then replace that NaN value with mean or median. It is preferred to impute using the median value as the average or the mean values are influenced by the outliers and skewness present in the data and are pulled in their respective direction.
* **Categorical Data:** When categorical data is missing, replace that with the value which is most occurring i.e. by mode.
* **Capping the data:**

We can place cap limits on the data again using three approaches. Oh yes! there are a lot of ways to deal with the data in machine learning 😀 So, can cap via:

* **Z-Score approach:**All the values above and below 3 standard deviations and are outliers and can be removed

#### Numerical data

The numerical data is scaled, meaning we bring all the numerical data on the same scale. For example, to predict how much loan amount to give to a customer depends on variables such as age, salary, number of working years. Now, on [***building a linear regression***](https://www.analytixlabs.co.in/blog/linear-regression-machine-learning/) model for this problem, it would not be possible for us to compare the beta coefficients of the above variables as the scale of each variable is different from the others. Hence, the Scaling of the variables is essential.  The two ways to scale data are Standardization and Normalization.

* **Standardization:** On the basis of the Z-score, the numerical data is scaled using the formula of calculating Z values =  (x-mean)/standard deviation. The data ranges in the interval of -3 to 3.
* **Normalization:**Here, the scaling happens using the formula: (x – min)/(max-min), reducing the data in the width of 0 to 1. This is also known as Min-Max Scalar.

#### 3.2.2 Categorical Data

The categorical data can not be directly fed into the model. We have seen machines are black and white, either 1 or 0. So, to use the categorical data for our model building process, we need to create dummy variables. Dummy variables are binary; they can take either the value as 1 or as 0. If we have n types of sub-categories within a categorical column, we must employ n-1 dummy variables. There are two ways to create dummy variables:

* Pandas’ function: pd.get\_dummies, and
* sklearn’s in-built function of OneHotEncoder

There is one more way of dealing with the categorical data, which is to use label encoding. The label encoder does not create dummy variables. However, it labels the categorical variable by numbers like below:

* Delhi   –>  1
* Mumbai   –>  2
* Hyderabad  –>  3

There is a limitation of label encoding: it converts the nominal data, which is the categorical data without any order, into ordinal data having order. In the above example, the three cities did not have order. However, the post applying label encoder has values 1,2,3, respectively. The machine will treat this data by giving precedence and treat the numbers as weights like 3 > 2 > 1 will make Hyderabad > Mumbai > Delhi. Hence, due to this limitation of label encoding, handling the categorical data is by creating the dummy variables.

## Data Preprocessing Steps in Machine Learning

The steps in data preprocessing in machine learning are:

1. Consolidation after [***acquisition of the data***](https://www.analytixlabs.co.in/blog/data-acquisition/)
2. Data Cleaning:
   1. Convert the data types if any mismatch present in the data types of the variables
   2. Change the format of the date variable to the required format
   3. Replace the special characters and constants with the appropriate values
3. Detection and treatment of missing values
4. Treating for negative values, if any present depending on the data
5. Outliers detection and treatment
6. Transformation of variables
7. Creation of new derived variables
8. Scale the numerical variables
9. Encode the categorical variables
10. Split the data into training, validation, and test set

**What is the key objective of data analysis?**

The primary objective of data analysis is to find meaningful insights within the data to use to make well-informed and accurate decisions.

**What is the difference between Scaling and Transformation?**

|  |  |  |
| --- | --- | --- |
|  | **Scaling** | **Transformation** |
| **Purpose** | The goal is to compare the variables as scaled variables on the same band can be compared and increase the computational power (or the efficiency). | Transformation helps in the case of skewed variables to reduce the skewness. In the case of regression, either if the assumptions of regression aren’t met or if the relationship between the target and independent variables is non-linear, then can use transformation to linearize. |
| **Impact on Data** | Scaling has no impact on the data. All the properties of the data remain the same—only the range of the independent variables changes. | Transformation changes the data, and so does the distribution of the data. |
| **Impact on Skewness, Kurtosis, Outliers** | As the distribution remains the same so no changes in skewness and kurtosis. Scaling doesn’t remove outliers. | Transformation can decrease the skewness. It brings values closer, which can remove the outliers. |

**What is the difference between Standardization and Normalization?**

Standardization and Normalization are scaling techniques. Standardization raises the data based on the Z-score, using the formula (x-mean)/standard deviation, reducing the data width to -3 to 3. Normalization scales the data using the formula (x – min)/(max-min) and Min-Max Scalar. It reduces the data width from 0 to 1.

**Steps Involved in Data Preprocessing:**

**1. Data Cleaning:**   
The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

* **(a). Missing Data:**   
  This situation arises when some data is missing in the data. It can be handled in various ways.   
  Some of them are:
  1. **Ignore the tuples:**   
     This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.
  2. **Fill the Missing values:**   
     There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.
* **(b). Noisy Data:**   
  Noisy data is a meaningless data that can’t be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :
  1. **Binning Method:**   
     This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.
  2. **Regression:**   
     Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).
  3. **Clustering:**   
     This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

**2. Data Transformation:**   
This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. **Normalization:**   
   It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)
2. **Attribute Selection:**   
   In this strategy, new attributes are constructed from the given set of attributes to help the mining process.
3. **Discretization:**   
   This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.
4. **Concept Hierarchy Generation:**   
   Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

**3. Data Reduction:**   
Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we uses data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.

The various steps to data reduction are:

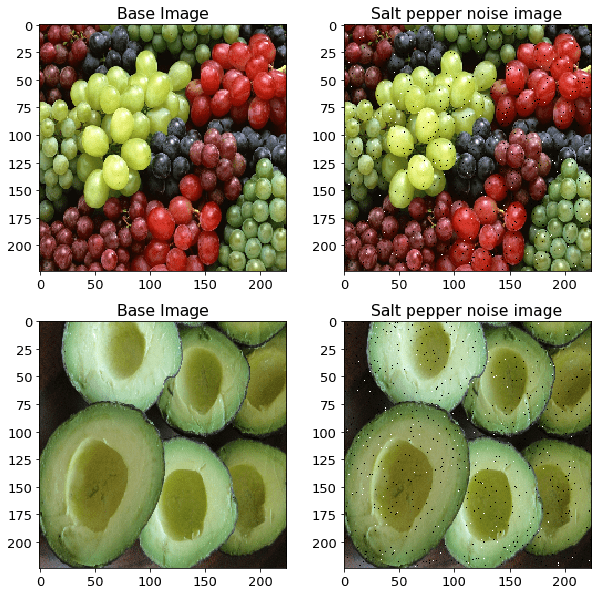
1. **Data Cube Aggregation:**   
   Aggregation operation is applied to data for the construction of the data cube.
2. **Attribute Subset Selection:**   
   The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value of the attribute.the attribute having p-value greater than significance level can be discarded.
3. **Numerosity Reduction:**   
   This enable to store the model of data instead of whole data, for example: Regression Models.
4. **Dimensionality Reduction:**   
   This reduce the size of data by encoding mechanisms.It can be lossy or lossless. If after reconstruction from compressed data, original data can be retrieved, such reduction are called lossless reduction else it is called lossy reduction. The two effective methods of dimensionality reduction are:Wavelet transforms and PCA (Principal Component Analysis).

## Data augmentation techniques in computer vision

There are geometric and color space augmentation methods for images to create image diversity in the model. It is easy to find many coding examples for these augmentation transformations from open source libraries and in articles on the topic.

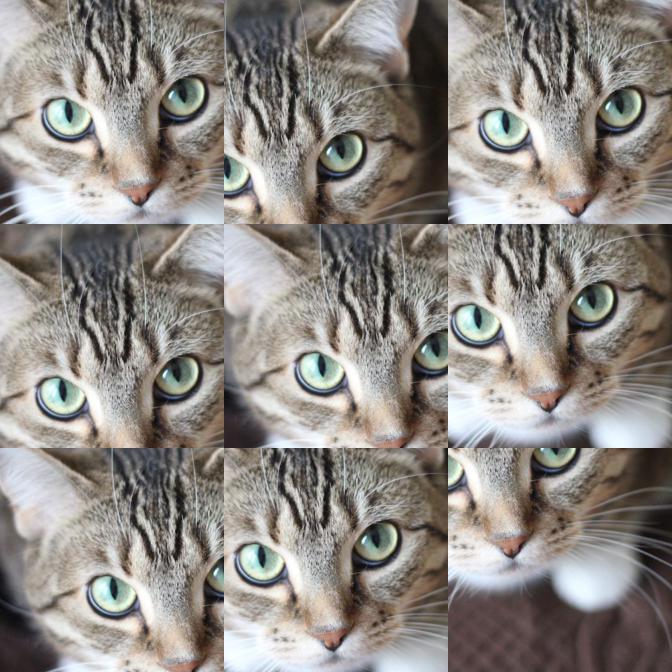
**Adding noise**

For blurry images, adding noise on the image can be useful. By “salt and pepper noise”, the image looks like consisting of white and black dots.



**Cropping**

A section of the image is selected, cropped and then resized to the original image size.

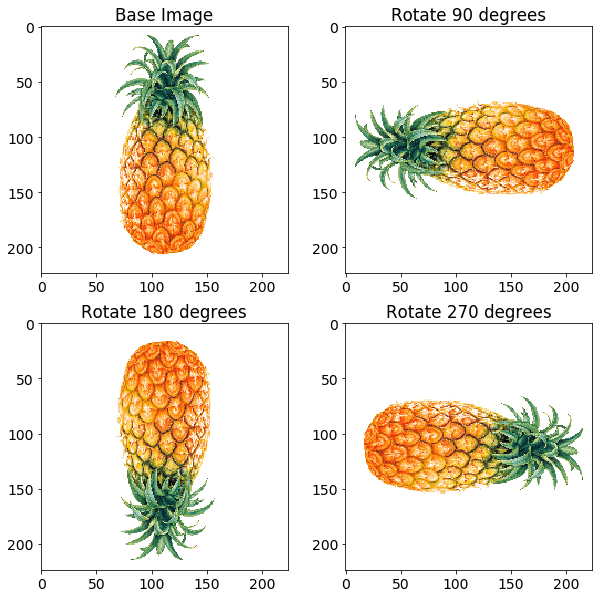


**Flipping**

The image is flipped horizontally and vertically. Flipping rearranges the pixels while protecting the features of the image. Vertical flipping is not meaningful for some photos, but it can be useful in cosmology or for microscopic photos.

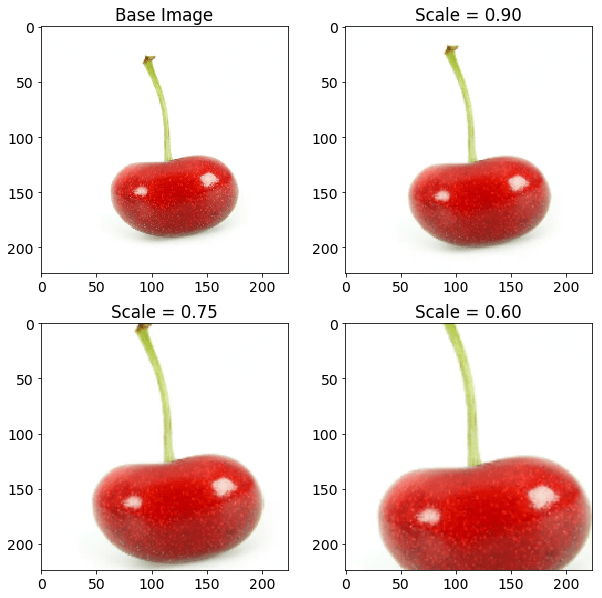


**Rotation**

The image is rotated by a degree between 0 and 360 degree. Every rotated image will be unique in the model.

**Scaling**

The image is scaled outward and inward. An object in new image can be smaller or bigger than in the original image by scaling



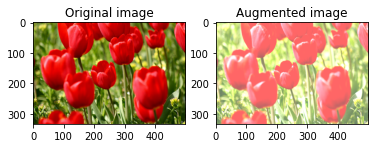
**Translation**

The image is shifted into various areas along the x-axis or y-axis, so neural network looks everywhere in the image to capture it.



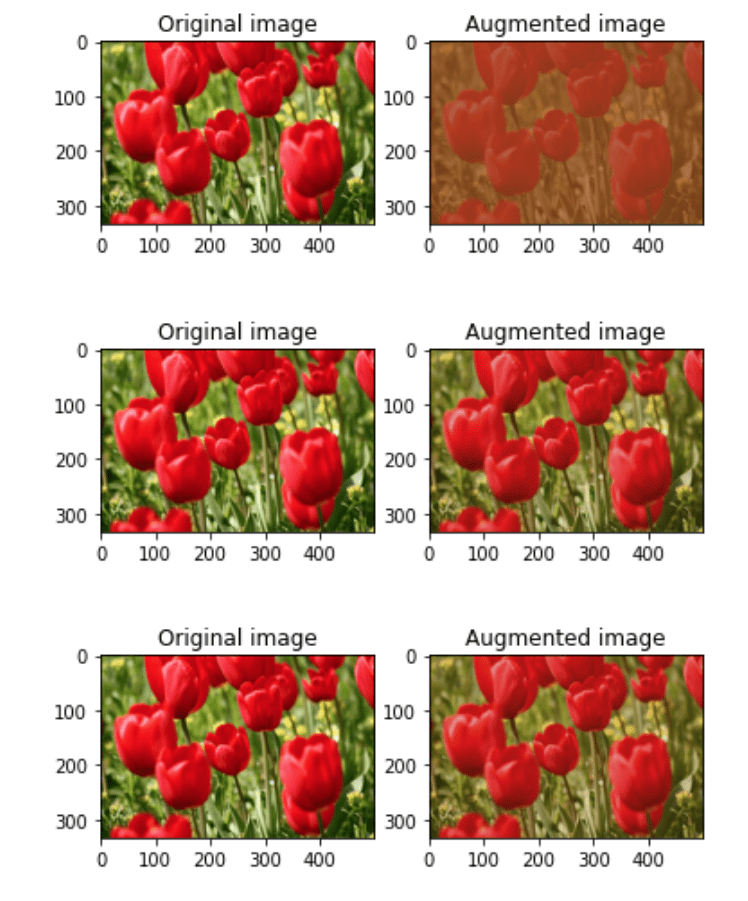
**Brightness**

The brightness of the image is changed and new image will be darker or lighter. This technique allows the model to recognize image in different lighting levels.



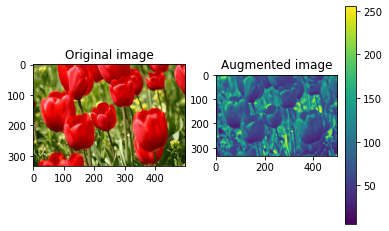
**Contrast**

The contrast of the image is changed and new image will be different from luminance and colour aspects. The following image’s contrast is changed randomly.



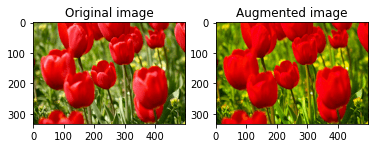
**Color Augmentation**

The color of image is changed by new pixel values. There is an example image which is grayscale.



**Saturation**

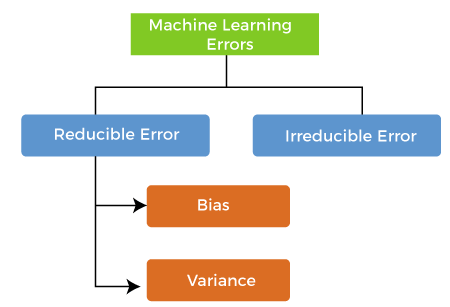
Saturation is depth or intensity of color in an image. The following image is saturated with data augmentation method.



## Errors in Machine Learning?

In machine learning, an error is a measure of how accurately an algorithm can make predictions for the previously unknown dataset. On the basis of these errors, the machine learning model is selected that can perform best on the particular dataset. There are mainly two types of errors in machine learning, which are:

* **Reducible errors:** These errors can be reduced to improve the model accuracy. Such errors can further be classified into bias and Variance.



* **Irreducible errors:** These errors will always be present in the model

regardless of which algorithm has been used. The cause of these errors is unknown variables whose value can't be reduced.

## What is Bias?

In general, a machine learning model analyses the data, find patterns in it and make predictions. While training, the model learns these patterns in the dataset and applies them to test data for prediction. **While making predictions, a difference occurs between prediction values made by the model and actual values/expected values**, **and this difference is known as bias errors or Errors due to bias**. It can be defined as an inability of machine learning algorithms such as Linear Regression to capture the true relationship between the data points. Each algorithm begins with some amount of bias because bias occurs from assumptions in the model, which makes the target function simple to learn. A model has either:

* **Low Bias:** A low bias model will make fewer assumptions about the form of the target function.
* **High Bias:** A model with a high bias makes more assumptions, and the model becomes unable to capture the important features of our dataset. **A high bias model also cannot perform well on new data.**

Generally, a linear algorithm has a high bias, as it makes them learn fast. The simpler the algorithm, the higher the bias it has likely to be introduced. Whereas a nonlinear algorithm often has low bias.

Some examples of machine learning algorithms with low bias **are Decision Trees, k-Nearest Neighbours and Support Vector Machines**. At the same time, an algorithm with high bias is **Linear Regression, Linear Discriminant Analysis and Logistic Regression.**

### Ways to reduce High Bias:

High bias mainly occurs due to a much simple model. Below are some ways to reduce the high bias:

* Increase the input features as the model is underfitted.
* Decrease the regularization term.
* Use more complex models, such as including some polynomial features.

## What is a Variance Error?

The variance would specify the amount of variation in the prediction if the different training data was used. In simple words, **variance tells that how much a random variable is different from its expected value**

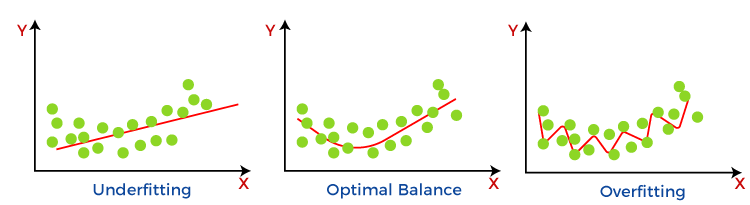
**Low variance** means there is a small variation in the prediction of the target function with changes in the training data set. At the same time, **High variance** shows a large variation in the prediction of the target function with changes in the training dataset.

A model that shows high variance learns a lot and perform well with the training dataset, and does not generalize well with the unseen dataset. As a result, such a model gives good results with the training dataset but shows high error rates on the test dataset.

Since, with high variance, the model learns too much from the dataset, it leads to overfitting of the model. A model with high variance has the below problems:

* A high variance model leads to overfitting.
* Increase model complexities.

Usually, nonlinear algorithms have a lot of flexibility to fit the model, have high variance.



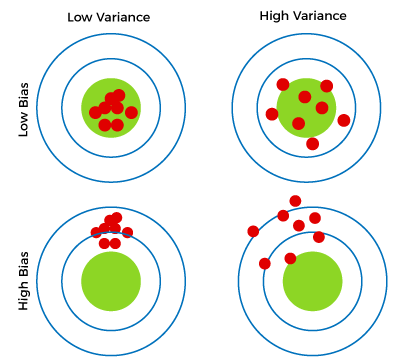
Some examples of machine learning algorithms with low variance are, **Linear Regression, Logistic Regression, and Linear discriminant analysis**. At the same time, algorithms with high variance are **decision tree, Support Vector Machine, and K-nearest neighbours.**

### Ways to Reduce High Variance:

* Reduce the input features or number of parameters as a model is overfitted.
* Do not use a much complex model.
* Increase the training data.
* Increase the Regularization term.

## Different Combinations of Bias-Variance

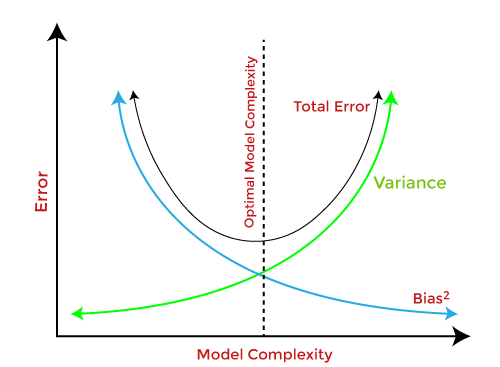
There are four possible combinations of bias and variances, which are represented by the below diagram



1. **Low-Bias, Low-Variance:**  
   The combination of low bias and low variance shows an ideal machine learning model. However, it is not possible practically.
2. **Low-Bias, High-Variance:** With low bias and high variance, model predictions are inconsistent and accurate on average. This case occurs when the model learns with a large number of parameters and hence leads to an **overfitting**
3. **High-Bias, Low-Variance:** With High bias and low variance, predictions are consistent but inaccurate on average. This case occurs when a model does not learn well with the training dataset or uses few numbers of the parameter. It leads to **underfitting** problems in the model.
4. **High-Bias, High-Variance:**  
   With high bias and high variance, predictions are inconsistent and also inaccurate on average.

## Bias-Variance Trade-Off

While building the machine learning model, it is really important to take care of bias and variance in order to avoid overfitting and underfitting in the model. If the model is very simple with fewer parameters, it may have low variance and high bias. Whereas, if the model has a large number of parameters, it will have high variance and low bias. So, it is required to make a balance between bias and variance errors, and this balance between the bias error and variance error is known as **the Bias-Variance trade-off.**

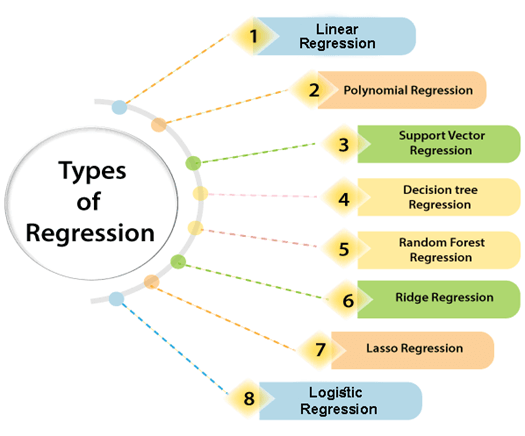


For an accurate prediction of the model, algorithms need a low variance and low bias. But this is not possible because bias and variance are related to each other:

* If we decrease the variance, it will increase the bias.
* If we decrease the bias, it will increase the variance.

## Terminologies Related to the Regression Analysis:

* **Dependent Variable:** The main factor in Regression analysis which we want to predict or understand is called the dependent variable. It is also called **target variable**.
* **Independent Variable:** The factors which affect the dependent variables or which are used to predict the values of the dependent variables are called independent variable, also called as a **predictor**.
* **Outliers:** Outlier is an observation which contains either very low value or very high value in comparison to other observed values. An outlier may hamper the result, so it should be avoided.
* **Multicollinearity:** If the independent variables are highly correlated with each other than other variables, then such condition is called Multicollinearity. It should not be present in the dataset, because it creates problem while ranking the most affecting variable.
* **Underfitting and Overfitting:** If our algorithm works well with the training dataset but not well with test dataset, then such problem is called **Overfitting**. And if our algorithm does not perform well even with training dataset, then such problem is called **underfitting**.



Linear Regression:

* Linear regression is a statistical regression method which is used for predictive analysis.
* It is one of the very simple and easy algorithms which works on regression and shows the relationship between the continuous variables.
* It is used for solving the regression problem in machine learning.
* Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), hence called linear regression.
* If there is only one input variable (x), then such linear regression is called **simple linear regression**. And if there is more than one input variable, then such linear regression is called **multiple linear regression**.
* The relationship between variables in the linear regression model can be explained using the below image. Here we are predicting the salary of an employee on the basis of **the year of experience**.
* Y= aX+b

Some popular applications of linear regression are:

* **Analyzing trends and sales estimates**
* **Salary forecasting**
* **Real estate prediction**
* **Arriving at ETAs in traffic.**

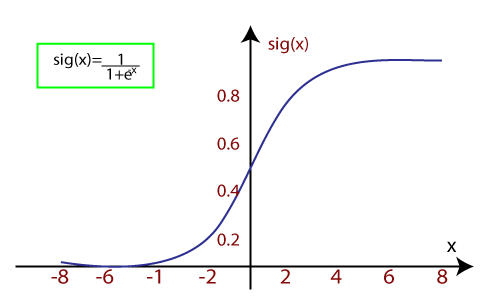
Logistic Regression:

* Logistic regression is another supervised learning algorithm which is used to solve the classification problems. In **classification problems**, we have dependent variables in a binary or discrete format such as 0 or 1.
* Logistic regression algorithm works with the categorical variable such as 0 or 1, Yes or No, True or False, Spam or not spam, etc.
* It is a predictive analysis algorithm which works on the concept of probability.
* Logistic regression is a type of regression, but it is different from the linear regression algorithm in the term how they are used.
* Logistic regression uses **sigmoid function** or logistic function which is a complex cost function. This sigmoid function is used to model the data in logistic regression. The function can be represented as:

Regression Analysis in Machine learning

* f(x)= Output between the 0 and 1 value.
* x= input to the function
* e= base of natural logarithm.

When we provide the input values (data) to the function, it gives the S-curve as follows:



Classification

Recall the different results from a binary classifier, which are true positives, true negatives, false positives, and false negatives. These are often shown in a confusion matrix. A confusion matrix is conceptually the basis of many classification performance metrics as shown. We will discuss a few of the more popular ones associated with machine learning here.

Accuracy is a key measure of performance, and is more specifically the rate at which the model is able to predict the correct value (classification or regression) for a given data point or observation. In other words, accuracy is the proportion of correct predictions out of all predictions made.

The other two metrics from the confusion matrix worth discussing are precision and recall. Precision (positive predictive value) is the ratio of true positives to the total amount of positive predictions made (i.e., true or false). Said another way, precision measures the proportion of accurate positive predictions out of all positive predictions made.

Recall on the other hand, or true positive rate, is the ratio of true positives to the total amount of actual positives, whether predicted correctly or not. So in other words, recall measures the proportion of accurate positive predictions out of all actual positive observations.

A metric that is associated with precision and recall is called the F-score (also called F1 score), which combines them mathematically, and somewhat like a weighted average, in order to produce a single measure of performance based on the simultaneous values of both. Its values range from 0 (worst) to 1 (best).

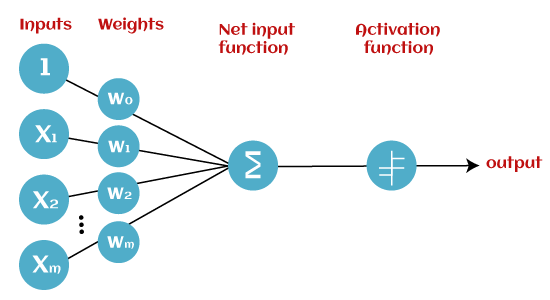
Another important concept to know about is the receiver operating characteristic, which when plotted, results in what’s known as an ROC curve (shown below, image courtesy of BOR at the English language Wikipedia).

An ROC curve is a two-dimensional plot of sensitivity (recall, or true positive rate) vs specificity (false positive rate). The area under the curve is referred to as the AUC, and is a numeric metric used to represent the quality and performance of the classifier (model). An AUC of 0.5 is essentially the same as random guessing without a model, whereas an AUC of 1.0 is considered a perfect classifier. Generally, the higher the AUC value the better, and an AUC above 0.8 is considered quite good.

The higher the AUC value, the closer the curve gets to the upper left corner of the plot. One can easily see from the ROC curves then that the goal is to find and tune a model that maximises the true positive rate, while simultaneously minimising the false positive rate. Said another way, the goal as shown by the ROC curve is to correctly predict as many of the actual positives as possible, while also predicting as many of the actual negatives as possible, and therefore minimise errors (incorrect classifications) for both.

## Basic Components of Perceptron

**Perceptron is also understood as an Artificial Neuron or neural network unit that helps to detect certain input data computations in business intelligence**.



* **Input Nodes or Input Layer:**

This is the primary component of Perceptron which accepts the initial data into the system for further processing. Each input node contains a real numerical value.

* **Wight and Bias:**

Weight parameter represents the strength of the connection between units. This is another most important parameter of Perceptron components. Weight is directly proportional to the strength of the associated input neuron in deciding the output. Further, Bias can be considered as the line of intercept in a linear equation.

* **Activation Function:**

These are the final and important components that help to determine whether the neuron will fire or not. Activation Function can be considered primarily as a step function.

Types of Activation functions:

* Sign function
* Step function, and
* Sigmoid function

## Simple Ensemble Techniques

In this section, we will look at a few simple but powerful techniques, namely:

1. Max Voting
2. Averaging
3. Weighted Averaging

#### 2.1 Max Voting

The max voting method is generally used for classification problems. In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a ‘vote’. The predictions which we get from the majority of the models are used as the final prediction.

For example, when you asked 5 of your colleagues to rate your movie (out of 5); we’ll assume three of them rated it as 4 while two of them gave it a 5. Since the majority gave a rating of 4, the final rating will be taken as 4. **You can consider this as taking the mode of all the predictions.**

The result of max voting would be something like this:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colleague 1 | Colleague 2 | Colleague 3 | Colleague 4 | Colleague 5 | Final rating |
| 5 | 4 | 5 | 4 | 4 | 4 |

#### 2.2 Averaging

Similar to the max voting technique, multiple predictions are made for each data point in averaging. In this method, we take an average of predictions from all the models and use it to make the final prediction. Averaging can be used for making predictions in regression problems or while calculating probabilities for classification problems.

For example, in the below case, the averaging method would take the average of all the values.

i.e. (5+4+5+4+4)/5 = 4.4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colleague 1 | Colleague 2 | Colleague 3 | Colleague 4 | Colleague 5 | Final rating |
| 5 | 4 | 5 | 4 | 4 | 4.4 |

#### 2.3 Weighted Average

This is an extension of the averaging method. All models are assigned different weights defining the importance of each model for prediction. For instance, if two of your colleagues are critics, while others have no prior experience in this field, then the answers by these two friends are given more importance as compared to the other people.

The result is calculated as [(5\*0.23) + (4\*0.23) + (5\*0.18) + (4\*0.18) + (4\*0.18)] = 4.41.

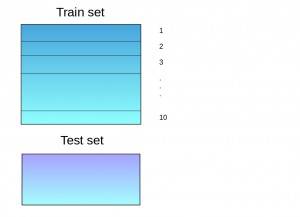
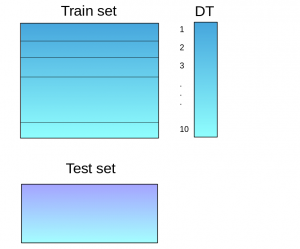
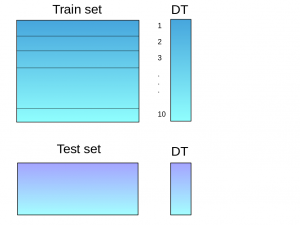
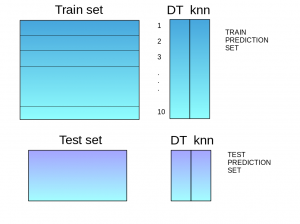
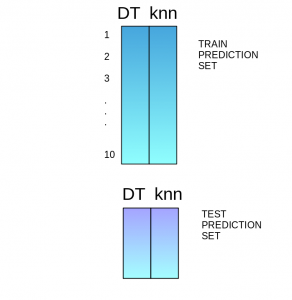
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colleague 1 | Colleague 2 | Colleague 3 | Colleague 4 | Colleague 5 | Final rating |
| weight | 0.23 | 0.23 | 0.18 | 0.18 | 0.18 |
| rating | 5 | 4 | 5 | 4 | 4 | 4.41 |

## 3. Advanced Ensemble techniques

Now that we have covered the basic ensemble techniques, let’s move on to understanding the advanced techniques.

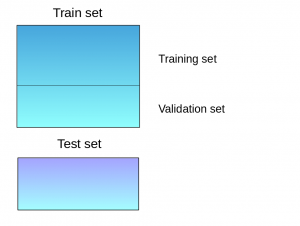
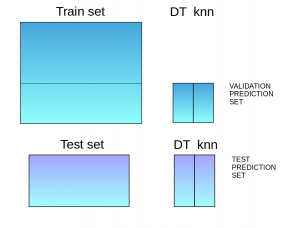
#### 3.1 Stacking

Stacking is an [ensemble learning](https://courses.analyticsvidhya.com/courses/ensemble-learning-and-ensemble-learning-techniques?utm_source=blog&utm_medium=comprehensive-guide-for-ensemble-models) technique that uses predictions from multiple models (for example decision tree, knn or svm) to build a new model. This model is used for making predictions on the test set. Below is a step-wise explanation for a simple stacked ensemble:

1. The train set is split into 10 parts.  
   
2. A base model (suppose a decision tree) is fitted on 9 parts and predictions are made for the 10th part. This is done for each part of the train set.  
   
3. The base model (in this case, decision tree) is then fitted on the whole train dataset.
4. Using this model, predictions are made on the test set.  
   
5. Steps 2 to 4 are repeated for another base model (say knn) resulting in another set of predictions for the train set and test set.  
   
6. The predictions from the train set are used as features to build a new model.  
   
7. This model is used to make final predictions on the test prediction set.

#### .2 Blending

Blending follows the same approach as stacking but uses only a holdout (validation) set from the train set to make predictions. In other words, unlike stacking, the predictions are made on the holdout set only. The holdout set and the predictions are used to build a model which is run on the test set. Here is a detailed explanation of the blending process:

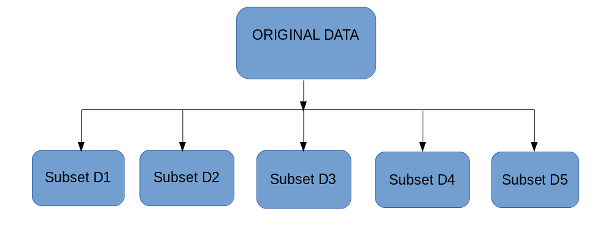
1. The train set is split into training and validation sets.  
   
2. Model(s) are fitted on the training set.
3. The predictions are made on the validation set and the test set.  
   
4. The validation set and its predictions are used as features to build a new model.
5. This model is used to make final predictions on the test and meta-features.

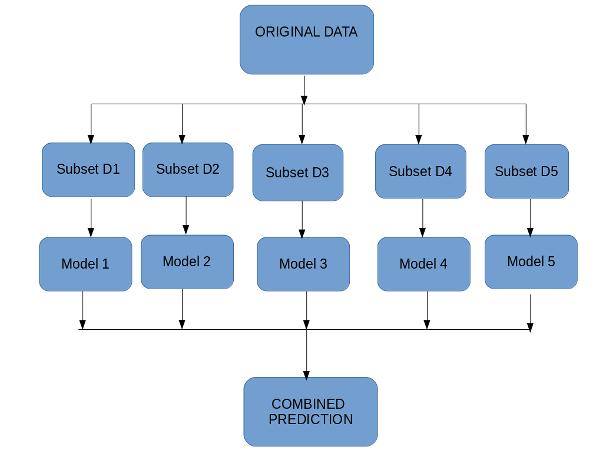
#### 3.3 Bagging

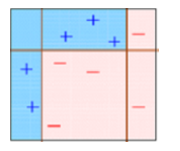
The idea behind bagging is combining the results of multiple models (for instance, all decision trees) to get a generalized result. Here’s a question: If you create all the models on the same set of data and combine it, will it be useful? There is a high chance that these models will give the same result since they are getting the same input. So how can we solve this problem? One of the techniques is bootstrapping.

Bootstrapping is a sampling technique in which we create subsets of observations from the original dataset, **with replacement**. The size of the subsets is the same as the size of the original set.

Bagging (or Bootstrap Aggregating) technique uses these subsets (bags) to get a fair idea of the distribution (complete set). The size of subsets created for bagging may be less than the original set.



1. Multiple subsets are created from the original dataset, selecting observations with replacement.
2. A base model (weak model) is created on each of these subsets.
3. The models run in parallel and are independent of each other.
4. The final predictions are determined by combining the predictions from all the models.



## 4. Algorithms based on Bagging and Boosting

Bagging and Boosting are two of the most commonly used techniques in machine learning. In this section, we will look at them in detail. Following are the algorithms we will be focusing on:

Bagging algorithms:

* Bagging meta-estimator
* Random forest

Boosting algorithms:

* AdaBoost
* GBM
* XGBM
* Light GBM
* CatBoost

For all the algorithms discussed in this section, we will follow this procedure:

* Introduction to the algorithm
* Sample code
* Parameters

For this article, I have used the Loan Prediction Problem. You can download the dataset from [**here**](https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/). Please note that a few code lines (reading the data, splitting into train-test sets, etc.)  will be the same for each algorithm. In order to avoid repetition, I have written the code for the same below, and further discussed only the code for the algorithm.

#importing important packages

import pandas as pd

import numpy as np

#reading the dataset

df=pd.read\_csv("/home/user/Desktop/train.csv")

#filling missing values

df['Gender'].fillna('Male', inplace=True)

#### 4.1 Bagging meta-estimator

Bagging meta-estimator is an ensembling algorithm that can be used for both classification (BaggingClassifier) and regression (BaggingRegressor) problems. It follows the typical bagging technique to make predictions. Following are the steps for the bagging meta-estimator algorithm:

1. Random subsets are created from the original dataset (Bootstrapping).
2. The subset of the dataset includes all features.
3. A user-specified base estimator is fitted on each of these smaller sets.
4. Predictions from each model are combined to get the final result.

**Code:**

from sklearn.ensemble import BaggingClassifier

from sklearn import tree

model = BaggingClassifier(tree.DecisionTreeClassifier(random\_state=1))

model.fit(x\_train, y\_train)

model.score(x\_test,y\_test)

0.75135135135135134

**Sample code for regression problem:**

from sklearn.ensemble import BaggingRegressor

model = BaggingRegressor(tree.DecisionTreeRegressor(random\_state=1))

model.fit(x\_train, y\_train)

model.score(x\_test,y\_test)

**Parameters used in the  algorithms:**

* **base\_estimator**:
  + It defines the base estimator to fit on random subsets of the dataset.
  + When nothing is specified, the base estimator is a decision tree.
* **n\_estimators**:
  + It is the number of base estimators to be created.
  + The number of estimators should be carefully tuned as a large number would take a very long time to run, while a very small number might not provide the best results.
* **max\_samples**:
  + This parameter controls the size of the subsets.
  + It is the maximum number of samples to train each base estimator.
* **max\_features**:
  + Controls the number of features to draw from the whole dataset.
  + It defines the maximum number of features required to train each base estimator.
* **n\_jobs**:
  + The number of jobs to run in parallel.
  + Set this value equal to the cores in your system.
  + If -1, the number of jobs is set to the number of cores.
* **random\_state**:
  + It specifies the method of random split. When random state value is same for two models, the random selection is same for both models.
  + This parameter is useful when you want to compare different models.

#### 4.2 Random Forest

Random Forest is another ensemble machine learning algorithm that follows the bagging technique. It is an extension of the bagging estimator algorithm. The base estimators in random forest are decision trees. Unlike bagging meta estimator, random forest randomly selects a set of features which are used to decide the best split at each node of the decision tree.

Looking at it step-by-step, this is what a random forest model does:

1. Random subsets are created from the original dataset (bootstrapping).
2. At each node in the decision tree, only a random set of features are considered to decide the best split.
3. A decision tree model is fitted on each of the subsets.
4. The final prediction is calculated by averaging the predictions from all decision trees.

*Note: The decision trees in random forest can be built on a subset of data and features. Particularly, the sklearn model of random forest uses all features for decision tree and a subset of features are randomly selected for splitting at each node.*

To sum up, Random forest **randomly**selects data points and features, and builds **multiple trees (Forest) .**  
**Code:**

**Parameters**

* **n\_estimators:**
  + It defines the number of decision trees to be created in a random forest.
  + Generally, a higher number makes the predictions stronger and more stable, but a very large number can result in higher training time.
* **criterion**:
  + It defines the function that is to be used for splitting.
  + The function measures the quality of a split for each feature and chooses the best split.
* **max\_features** :
  + It defines the maximum number of features allowed for the split in each decision tree.
  + Increasing max features usually improve performance but a very high number can decrease the diversity of each tree.
* **max\_depth**:
  + Random forest has multiple decision trees. This parameter defines the maximum depth of the trees.
* **min\_samples\_split:**
  + Used to define the minimum number of samples required in a leaf node before a split is attempted.
  + If the number of samples is less than the required number, the node is not split.
* **min\_samples\_leaf:**
  + This defines the minimum number of samples required to be at a leaf node.
  + Smaller leaf size makes the model more prone to capturing noise in train data.
* **max\_leaf\_nodes:**
  + This parameter specifies the maximum number of leaf nodes for each tree.
  + The tree stops splitting when the number of leaf nodes becomes equal to the max leaf node.
* **n\_jobs**:
  + This indicates the number of jobs to run in parallel.
  + Set value to -1 if you want it to run on all cores in the system.
* **random\_state**:
  + This parameter is used to define the random selection.
  + It is used for comparison between various models.

#### 4.3 AdaBoost

Adaptive boosting or AdaBoost is one of the simplest boosting algorithms. Usually, decision trees are used for modelling. Multiple sequential models are created, each correcting the errors from the last model. AdaBoost assigns weights to the observations which are incorrectly predicted and the subsequent model works to predict these values correctly.

Below are the steps for performing the AdaBoost algorithm:

1. Initially, all observations in the dataset are given equal weights.
2. A model is built on a subset of data.
3. Using this model, predictions are made on the whole dataset.
4. Errors are calculated by comparing the predictions and actual values.
5. While creating the next model, higher weights are given to the data points which were predicted incorrectly.
6. Weights can be determined using the error value. For instance, higher the error more is the weight assigned to the observation.
7. This process is repeated until the error function does not change, or the maximum limit of the number of estimators is reached.

**Code:**

from sklearn.ensemble import AdaBoostClassifier

model = AdaBoostClassifier(random\_state=1)

model.fit(x\_train, y\_train)

model.score(x\_test,y\_test)

0.81081081081081086

**Sample code for regression problem:**

from sklearn.ensemble import AdaBoostRegressor

model = AdaBoostRegressor()

model.fit(x\_train, y\_train)

model.score(x\_test,y\_test)

**Parameters**

* **base\_estimators**:
  + It helps to specify the type of base estimator, that is, the machine learning algorithm to be used as base learner.
* **n\_estimators:**
  + It defines the number of base estimators.
  + The default value is 10, but you should keep a higher value to get better performance.
* **learning\_rate:**
  + This parameter controls the contribution of the estimators in the final combination.
  + There is a trade-off between learning\_rate and n\_estimators.
* **max\_depth**:
  + Defines the maximum depth of the individual estimator.
  + Tune this parameter for best performance.
* **n\_jobs**
  + Specifies the number of processors it is allowed to use.
  + Set value to -1 for maximum processors allowed.
* **random\_state** :
  + An integer value to specify the random data split.
  + A definite value of random\_state will always produce same results if given with same parameters and training data.

#### 4.4 Gradient Boosting (GBM)

Gradient Boosting or GBM is another ensemble machine learning algorithm that works for both regression and classification problems. GBM uses the boosting technique, combining a number of weak learners to form a strong learner. Regression trees used as a base learner, each subsequent tree in series is built on the errors calculated by the previous tree.

**Differences Between Bagging and Boosting**

| **S.NO** | **Bagging** | **Boosting** |
| --- | --- | --- |
| **1.** | **The simplest way of combining predictions that  belong to the same type.** | **A way of combining predictions that  belong to the different types.** |
| **2.** | **Aim to decrease variance, not bias.** | **Aim to decrease bias, not variance.** |
| **3.** | **Each model receives equal weight.** | **Models are weighted according to their performance.** |
| **4.** | **Each model is built independently.** | **New models are influenced  by the performance of previously built models.** |
|  |  |  |
| **5.** | **Different training data subsets are selected using row sampling with replacement and random sampling methods from the entire training dataset.** | **Every new subset contains the elements that were misclassified by previous models.** |
| **6.** | **Bagging tries to solve the over-fitting problem.** | **Boosting tries to reduce bias.** |
| **7.** | **If the classifier is unstable (high variance), then apply bagging.** | **If the classifier is stable and simple (high bias) the apply boosting.** |
| **8.** | **In this base classifiers are trained parallelly.** | **In this base classifiers are trained sequentially.** |
| **9** | **Example: The Random forest model uses Bagging.** | **Example: The AdaBoost uses Boosting techniques** |

**Bagging**: It is a homogeneous weak learners’ model that learns from each other independently in parallel and combines them for determining the model average.

**Boosting**: It is also a homogeneous weak learners’ model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

## Bagging

**B**ootstrap **A**ggregating, also known as bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It decreases the [variance](https://www.geeksforgeeks.org/mathematics-mean-variance-and-standard-deviation/)and helps to avoid [overfitting](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/). It is usually applied to [decision tree methods](https://www.geeksforgeeks.org/decision-tree/). Bagging is a special case of the model averaging approach.

**Implementation Steps of Bagging**

* **Step 1:** Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
* **Step 2:** A base model is created on each of these subsets.
* **Step 3:**Each model is learned in parallel with each training set and independent of each other.
* **Step 4:**The final predictions are determined by combining the predictions from all the models.

## Boosting

Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models is added

**Algorithm**

1. *Initialise the dataset and assign equal weight to each of the data point.*
2. *Provide this as input to the model and identify the wrongly classified data points.*
3. *Increase the weight of the wrongly classified data points and decrease the weights of correctly classified data points. And then normalize the weights of all data points.*
4. *if (got required results)  
     Goto step 5  
   else  
     Goto step 2*
5. *End*

**Similarities Between Bagging and Boosting**

Bagging and Boosting, both being the commonly used methods, have a universal similarity of being classified as ensemble methods. Here we will explain the similarities between them.

1. Both are ensemble methods to get N learners from 1 learner.
2. Both generate several training data sets by random sampling.
3. Both make the final decision by averaging the N learners (or taking the majority of them i.e Majority Voting).
4. Both are good at reducing variance and provide higher stability.