## What is supervised learning?

[Supervised learning](https://www.ibm.com/cloud/learn/supervised-learning) is a machine learning approach that’s defined by its use of labeled datasets. These datasets are designed to train or “supervise” algorithms into classifying data or predicting outcomes accurately. Using labeled inputs and outputs, the model can measure its accuracy and learn over time.

* **Classification** problems use an algorithm to accurately assign test data into specific categories, such as separating apples from oranges. Or, in the real world, supervised learning algorithms can be used to classify spam in a separate folder from your inbox. Linear classifiers, support vector machines, decision trees and [random forest](https://www.ibm.com/cloud/learn/random-forest) are all common types of classification algorithms.
* **Regression** is another type of supervised learning method that uses an algorithm to understand the relationship between dependent and independent variables. Regression models are helpful for predicting numerical values based on different data points, such as sales revenue projections for a given business. Some popular regression algorithms are linear regression, logistic regression and polynomial regression.

## What is unsupervised learning?

[Unsupervised learning](https://www.ibm.com/cloud/learn/unsupervised-learning) uses machine learning algorithms to analyze and cluster unlabeled data sets. These algorithms discover hidden patterns in data without the need for human intervention (hence, they are “unsupervised”).

Unsupervised learning models are used for three main tasks: clustering, association and dimensionality reduction:

* **Clustering** is a data mining technique for grouping unlabeled data based on their similarities or differences. For example, K-means clustering algorithms assign similar data points into groups, where the K value represents the size of the grouping and granularity. This technique is helpful for market segmentation, image compression, etc.
* **Association** is another type of unsupervised learning method that uses different rules to find relationships between variables in a given dataset. These methods are frequently used for market basket analysis and recommendation engines, along the lines of “Customers Who Bought This Item Also Bought” recommendations.
* **Dimensionality reduction** is a learning technique used when the number of features  (or dimensions) in a given dataset is too high. It reduces the number of data inputs to a manageable size while also preserving the data integrity. Often, this technique is used in the preprocessing data stage, such as when autoencoders remove noise from visual data to improve picture quality.

In [statistics](https://www.scribbr.com/?cat_ID=34372), a **Type I error** is a false positive conclusion, while a **Type II error** is a false negative conclusion.

Making a statistical decision always involves uncertainties, so the risks of making these errors are unavoidable in [hypothesis testing](https://www.scribbr.com/statistics/hypothesis-testing/).

The probability of making a Type I error is the significance level, or alpha (α), while the probability of making a Type II error is beta (β). These risks can be minimized through careful planning in your study design.

Example: Type I vs Type II errorYou decide to get tested for COVID-19 based on mild symptoms. There are two errors that could potentially occur:

* **Type I error (false positive):** the test result says you have coronavirus, but you actually don’t.
* **Type II error (false negative):** the test result says you don’t have coronavirus, but you actually do.

Pruning in [Artificial Intelligence](https://intellipaat.com/blog/what-is-artificial-intelligence/) is removing the nodes from the model to reach a better solution. Pruning is blocking the leaf nodes and removing the entire sub-tree to increase prediction accuracy by reduces the overfitting. Alpha-beta pruning that is similar to the min-max algorithm is the most used pruning algorithm in Artificial intelligence.

Pruning is the process of removing weight connections in a network to increase inference speed and decrease model storage size. In general, neural networks are very over parameterized. Pruning a network can be thought of as removing unused parameters from the over parameterized network.

Advantages and Disadvantages of KNN Algorithm

KNN is a very simple algorithm used to solve classification problems. KNN stands for K-Nearest Neighbors. K is the number of neighbors in KNN. Lets find out some advantages and disadvantages of KNN algorithm.  
  
**Advantages of KNN**  
  
**1. No Training Period:** KNN is called **Lazy Learner (Instance based learning)**. It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.  
  
**2.** Since the KNN algorithm requires no training before making predictions, **new data can be added seamlessly** which will not impact the accuracy of the algorithm.  
  
**3.** KNN is very **easy to implement**. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)  
  
**Disadvantages of KNN**  
  
**1. Does not work well with large dataset:**In large datasets, the cost of calculating the distance between the new point and each existing points is huge which degrades the performance of the algorithm.  
  
**2. Does not work well with high dimensions:**The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension.  
  
**3. Need feature scaling:** We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions.  
  
**4. Sensitive to noisy data, missing values and outliers**: KNN is sensitive to noise in the dataset. We need to manually impute missing values and remove outliers.

Naïve Bayes

* Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
* It is mainly used in *text classification* that includes a high-dimensional training dataset.
* Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
* **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

* **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
* **Bayes**: It is called Bayes because it depends on the principle of [Bayes' Theorem](https://www.javatpoint.com/bayes-theorem-in-artifical-intelligence).

Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

Suppose we have a dataset of **weather conditions** and corresponding target variable "**Play**". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

AdaBoost and Gradient Boost

AdaBoost is the first designed boosting algorithm with a particular loss function. On the other hand, Gradient Boosting is a generic algorithm that assists in searching the approximate solutions to the additive modelling problem. This makes Gradient Boosting more flexible than AdaBoost.

### **AdaBoost**

AdaBoost or Adaptive Boosting is the first [Boosting ensemble model](https://analyticsindiamag.com/ensemble-modeling-explained-through-music/). The method automatically adjusts its parameters to the data based on the actual performance in the current iteration. Meaning, both the weights for re-weighting the data and the weights for the final aggregation are re-computed iteratively.

In practice, this boosting technique is used with [simple classification trees](https://arxiv.org/pdf/1403.1452.pdf) or stumps as base-learners, which resulted in improved performance compared to the classification by one tree or other single base-learner.

The most common algorithm used with AdaBoost is **decision trees with one level that means with Decision trees with only 1 split**.

**Ada Boost 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.*
4. *if (got required results)   
     Goto step 5   
   else   
     Goto step 2*
5. *End*

Gradient Boost

Gradient boost algorithm *helps us minimize bias error* of the model. This algorithm starts by building a decision stump and then assigning equal weights to all the data points. Then it increases the weights for all the points which are misclassified and lowers the weight for those that are easy to classify or are correctly classified. A new decision stump is made for these weighted data points. The idea behind this is to improve the predictions made by the first stump.

Voting Classifier

A voting classifier is **a machine learning estimator that trains various base models or estimators and predicts on the basis of aggregating the findings of each base estimator**. The aggregating criteria can be combined decision of voting for each estimator output.

Two different voting schemes are common among voting classifiers:

In **hard voting** (also known as majority voting), every individual classifier votes for a class, and the majority wins. In statistical terms, the predicted target label of the ensemble is the mode of the distribution of individually predicted labels.

In soft voting, every individual classifier provides a probability value that a specific data point belongs to a particular target class. The predictions are weighted by the classifier's importance and summed up. Then the target label with the greatest sum of weighted probabilities wins the vote.

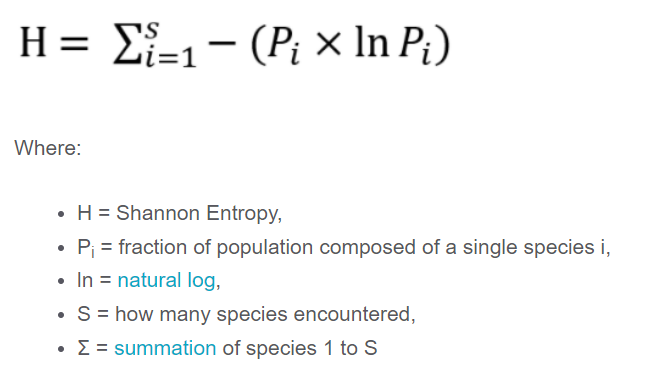
Stacking Classifier

Stacking is one of the most popular ensemble machine learning techniques used to predict multiple nodes to build a new model and improve model performance. Stacking enables us to train multiple models to solve similar problems, and based on their combined output, it builds a new model with improved performance.

Stacking is a way of ensembling classification or regression models it consists of two-layer estimators. The first layer consists of all the baseline models that are used to predict the outputs on the test datasets. The second layer consists of Meta-Classifier or Regressor which takes all the predictions of baseline models as an input and generate new predictions.

**Shannons Entropy and Conditional Entropy**

The “Shannon entropy” is a concept introduced by Shannon (1948), where **a measure of the uncertainty of occurrence of certain event, given partial information about the system, is proposed**. **Shannon entropy** (or just entropy) is a measure of [uncertainty](https://www.statisticshowto.com/uncertainty-in-statistics/) (or [variability](https://www.statisticshowto.com/variability/)) associated with [random variables](https://www.statisticshowto.com/random-variable/).

****

**Conditional Entropy**

The conditional entropy measures how much entropy a random variable X has remaining if we have already learned the value of a second random variable Y. It is referred to as the entropy of X conditional on Y, and is written H(X∣Y).

**Define precison and recall with examples.**

We can calculate the precision as follows:

Precision = TruePositives / (TruePositives +FalsePositives)

We can calculate the recall as follows:

Recall = TruePositives / (TruePositives + FalseNegatives)

Precision and recall are two extremely important model evaluation metrics. While precision refers to the percentage of your results which are relevant, recall refers to the percentage of total relevant results correctly classified by your algorithm.

**Define pruning and when do you use it.**

Pruning is a data compression technique in machine learning and search algorithms that reduces the size of decision trees by removing sections of the tree that are non-critical and redundant to classify instances. Pruning in Artificial Intelligence is removing the nodes from the model to reach a better solution. Pruning is blocking the leaf nodes and removing the entire sub-tree to increase prediction accuracy by reducing the overfitting

pre-pruning or early stopping involves stopping the tree before it has completed classifying the training set and post-pruning refers to pruning the tree after it has finished.