Machine Learning:

"Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed." —Arthur Samuel, 1959

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

In easy words we can defined Machine learning (ML) as a part of artificial intelligence (AI) that uses data and algorithms to help computers learn and get better at tasks much like how humans learn from their experience.

[Reference: https://www.ibm.com,

Book: Hands-On-Machine-Learning by Aurelien Geron]

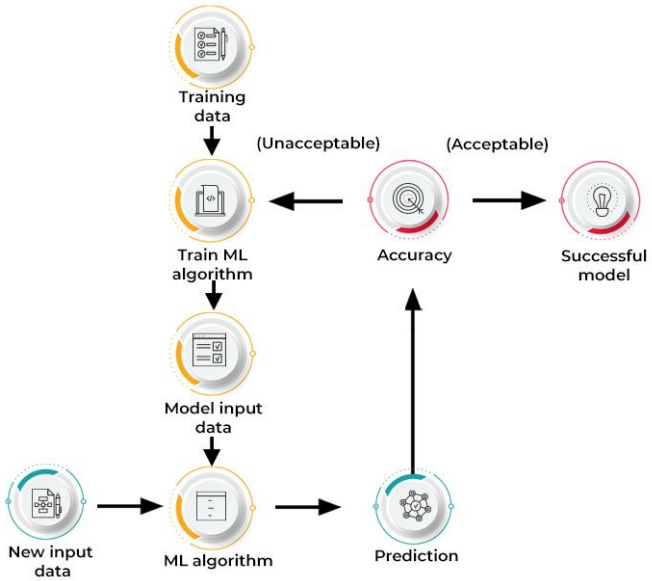
The key elements of machine learning:

There are three main elements to every machine learning algorithm and they include:

- Representation (what the model looks like, how knowledge is represented)
- Evaluation: (how programs are evaluated)
- Optimization: (how programs are generated)

[Reference: https://www.domo.com]

How machine learning works:



[Img Source: https://www.spiceworks.com]

Description:

- The process start with Training Data(It's like studying notes before an exam)
- Train the Machine Learning(ML) algorithm using these data (This is like practicing problems to understand concepts better)
- Test the ML algorithm with some input data (like a practice test)
- Use new data on ML algorithm(the actual exam, where student who studied are now ready for the exam)
- Make Predictions on new data (answers written on the exam)
- Check Accuracy of these predictions (grading the exam to see if the answers are right)
- If accuracy is acceptable, the model is successful (passing the exam with good grades)
- If accuracy is unacceptable, retrain the ML algorithm (need to study more if the exam grades are poor)

This loop continues until the model is accurate enough to be used in real-world scenarios.

Types of machine learning: Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic types of machine learning:

1. Supervised machine learning:

Supervised machine learning is defined by its use of labeled datasets to train algorithms to classify data or predict outcomes accurately. Within supervised learning various algorithms are used including: Linear regression, Decision trees etc.

Example: Consider an input dataset of parrot and crow images. Initially, the machine is trained to understand the pictures, including the parrot and crow's color, eyes, shape, and size. During post-training an input picture of a parrot is provided and the machine is expected to identify the object and predict the output. The trained machine checks for the various features of the object such as color, eyes, shape, etc. in the input picture to make a final prediction. This is the process of object identification in supervised machine learning.

Example: A supervised model might be used to predict flight times based on specific parameters, such as weather conditions, airport traffic, peak flight hours, and more.

2. Unsupervised machine learning:

Unsupervised learning refers to a learning technique that's devoid of supervision. Here, the machine is trained using an unlabeled dataset and is enabled to predict the output without any supervision. The common type of algorithm used in unsupervised learning is K-Means or clustering.

Example: Consider an input dataset of images of a fruit-filled container. Here, the images are not known to the machine learning model. When we input the dataset into the ML model, the task of the model is to identify the pattern of objects, such as color, shape in the input images and categorize them. Upon categorization, the machine then predicts the output as it gets tested with a test dataset.

Example: Anomaly detection: Unsupervised clustering can process large datasets and discover data point that are atypical in a dataset.

3. Semi-supervised learning:

Semi-supervised learning comprises characteristics of both supervised and unsupervised machine learning. It uses the combination of labeled and unlabeled datasets to train its algorithms. Semi-supervised learning can be used in Machine translation, Fraud detection etc.

Example: Consider a college student learning a concept under a teacher's supervision in college is termed supervised learning. In unsupervised learning, a student self-learns the same concept at home without a teacher's guidance. Meanwhile, a student revising the concept after learning under the direction of a teacher in college is a semi-supervised form of learning.

4. Reinforcement learning:

Reinforcement learning is a feedback-based process. Reinforcement learning works by programming an algorithm with a distinct goal and a prescribed set of rules for accomplishing that goal. Unlike supervised learning, reinforcement learning lacks labeled data, and the agents learn via experiences only. Reinforcement learning is applied across different fields such as game theory, information theory.

Example: Consider video games. Here, the game specifies the environment, and each move of the reinforcement agent defines its state. The agent is entitled to receive feedback via punishment and rewards, thereby affecting the overall game score. The ultimate goal of the agent is to achieve a high score.

[Reference: https://www.ibm.com/topics/machine-learning
https://www.techtarget.com , https://cloud.google.com ,
https://www.spiceworks.com , https://www.domo.com]

Classification & Regression:

Supervised learning includes regression and classification. Regression is when the variable to predict is numerical whereas classification is when the variable to predict is categorical.

Features	Regression	Classification
Main goal	Predicts continuous values like salary and age.	Predicts discrete values like stock and forecasts.
Input and output variables	Input: Either categorical or continuousOutput: Only continuous	Input: Either categorical or continuousOutput: Only categorial
Types of algorithm	Linear regressionPolynomial regressionLasso regressionRidge regression	Decision treesRandom forestsLogistic regressionNeural networksSupport vector machines
Evaluation metric	R2 scoreMean squared errorMean absolute errorAbsolute percentage error (MAPE)	Receiver operating characteristic curveRecallAccuracyPrecisionF1 score

[Img Source: https://www.analyticsvidhya.com/blog/2023/05/regression-vs-classification]

Classification:

These refer to algorithms that address classification problems where the output variable is categorical. For example, yes or no, true or false, male or female etc. Real-world applications of this category are evident in spam detection and email filtering. Some known classification algorithms include the Random Forest Algorithm, Decision Tree Algorithm, Logistic Regression Algorithm etc.

Regression:

Regression algorithms handle regression problems where input and output variables have a linear relationship. These are known to predict continuous output variables. Examples include weather prediction, market trend analysis, etc. Popular regression algorithms include the Simple Linear Regression Algorithm, Multivariate Regression Algorithm, Decision Tree Algorithm etc.

[Reference: https://www.spiceworks.com/ , https://www.domo.com]

Dataset:

A dataset in machine learning refers to a collection of data that is used to train and test algorithms and models. A dataset is typically divided into three subsets: training set, validation set, and test set.

1. Training Dataset:

The training set is the largest portion of the dataset portion of the dataset reserved to fit the model. In other words, the model sees and learns from the data in the training set to directly improve its parameters.

Example: In a dataset of house prices, the training set includes features like square footage, number of bedrooms, and location, along with the corresponding house prices.

2. Validation Dataset: The validation set is the set of data used to evaluate and fine-tune a machine learning model during training, helping to assess the model's performance and make adjustments. By evaluating a trained model on the validation set, we gain insights into its ability to generalize to unseen data.

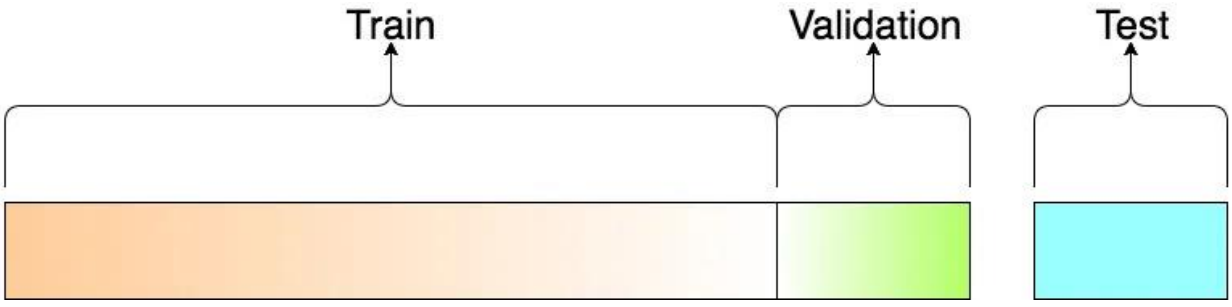
Example: Using the same house price dataset, the validation set might contain a smaller, separate subset of the data that the model has not seen during training. This set helps determine if the model is learning correctly or if it is starting to over-fit the training data.

3. Test Dataset: The test set is the set of data used to evaluate the final performance of a trained model. The Test dataset provides the gold standard used to evaluate the model. It is only used once a model is completely trained.

Example: The test set in the house price dataset includes data points that were not used in either the training or validation sets. The final performance metrics on this set, such as mean absolute error or root mean squared error, indicate how well the model is likely to perform in real-world applications.

The train-test-validation split is fundamental in machine learning, particularly during model development. A common practice is to split the dataset (100%) into 70-80% for training, 10-15% for validation, and 10-15% for testing. The exact split can vary depending on the size of the dataset and the specific needs of the project.

[Reference: https://encord.com , https://www.analyticsvidhya.com]



A visualization of the splits dataset into Train, Validation and Test sets.

[Img Source: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7]

Epoch and Batch in Dataset:

Epoch:

An epoch is when all the training data is used at once and is defined as the total number of iterations of all the training data in one cycle for training the machine learning model. Another way to define an epoch is the number of passes a training dataset takes around an algorithm. One pass is counted when the data set has done both forward and backward passes. The number of epochs is considered a hyperparameter.

Example: If a dataset has 1000 samples and the model is trained for 10 epochs, the model will see each of the 1000 samples 10 times over the course of training.

Batch:

The batch is the dataset that has been divided into smaller parts to be fed into the algorithm.

A training dataset can be broken down into multiple batches. If only a single batch exists, then the learning algorithm is called batch gradient descent. The learning algorithm is called stochastic gradient descent, when an entire sample makes up a batch. The algorithm is called a mini-batch gradient descent when the batch size is more than one sample but less than the training dataset size.

Example: If the dataset has 1000 samples and the batch size is 100, the dataset will be divided into 10 batches. Training on the entire dataset constitutes 1 epoch, which will have 10 iterations (one for each batch).

[Reference: https://www.simplilearn.com]

Feature in Machine Learning: A feature is an individual measurable property within a recorded dataset. Features can also be defined as individual measurable properties or characteristics of the data used in modeling. In machine learning and statistics, features are often called variables or attributes.

Example: In a patient medical dataset, features could be age, gender, blood pressure, cholesterol level, and other observed characteristics relevant to the patient.

The type of feature that is used in feature engineering depends on the specific machine learning algorithm that is being used. In feature engineering, two types of features are commonly used: numerical and categorical.

1. **Numerical features**: These are continuous values that can be measured on a scale. Examples of numerical features include age, height, weight, and income. Numerical features can be used in machine learning algorithms directly.

2. **Categorical features:** These are discrete values that can be grouped into categories. Examples of categorical features include gender, color, and zip code. Categorical features typically need to be converted to numerical features before they can be used in machine learning algorithms.

[Reference: https://domino.ai , https://en.wikipedia.org]

Generative Model:

A generative model is a type of machine learning model that aims to learn the underlying patterns or distributions of data in order to generate new, similar data. These models focus on understanding how the data is generated. They aim to learn the distribution of the data itself. Few types of generative models are diffusion models, Markov chains, GNNs (Generative Adversarial Networks) etc. Applications of these models can be in art creation, content creation, drug discovery etc.

Example: If we're looking at pictures of cats and dogs, a generative model would try to understand what makes a cat look like a cat and a dog look like a dog. It would then be able to generate new images that resemble either cats or dogs.

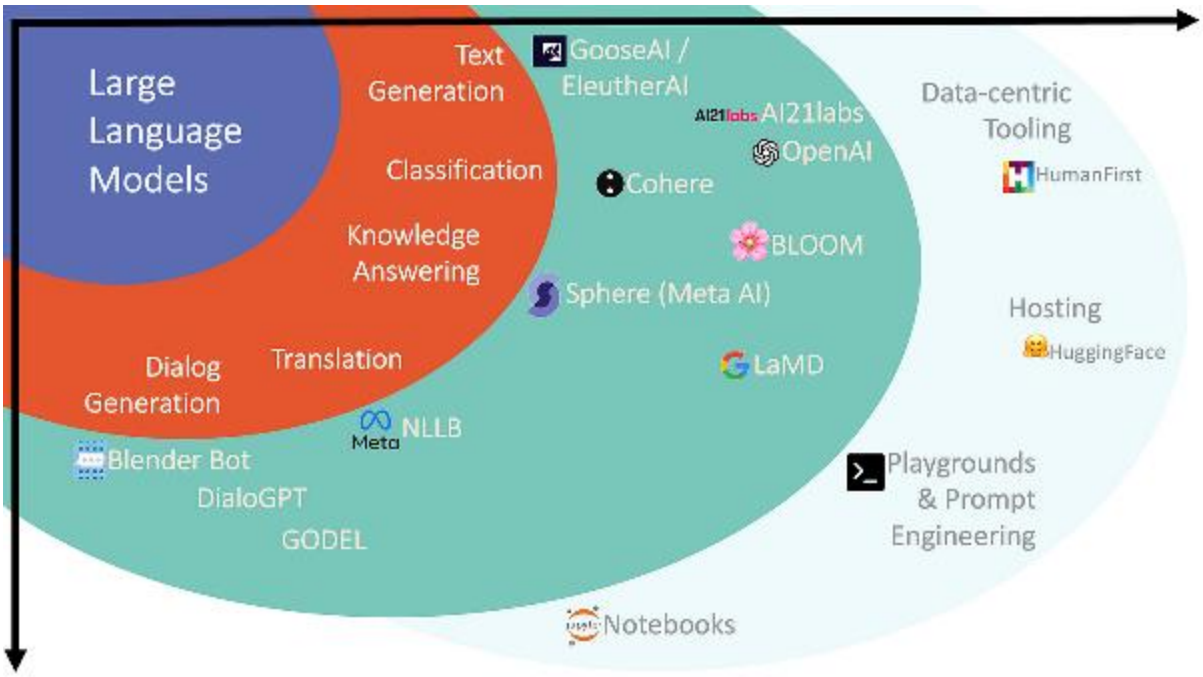
[Reference: https://www.datacamp.com]

Large Language Models (LLMs):

A large language model is a type of artificial intelligence algorithm that uses deep learning techniques and massively large data sets to understand, summarize, generate and predict new content. The term generative AI also is closely connected with LLMs. Generative AI can generate text, code, images, video, and music, whereas large language models are a type of generative AI that are trained on text and produce textual content. Large language models are also referred to as neural networks (NNs), which are computing systems inspired by the human brain. These neural networks work using a network of nodes that are layered, much like neurons.

Few examples of LLMs are Bidirectional Encoder Representations from Transformers (Bert), Falcon 40B, GPT-3, GPT-3.5, GPT-4 etc.

[Reference: https://www.techtarget.com , https://www.elastic.com]



Evaluate a Large Language Model

[Img source: https://www.analyticsvidhya.com]

Artificial Neural Networks:

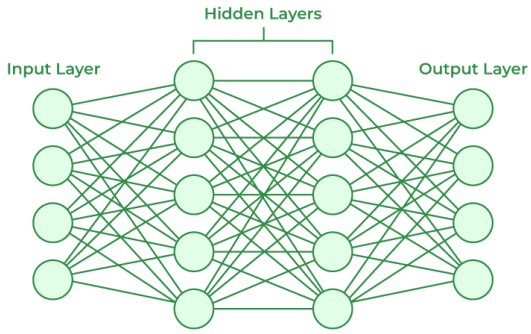
The term "Artificial Neural Network" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

An Artificial Neural Network in the field of Artificial intelligence where it attempts to mimic the network of neurons makes up a human brain so that computers will have an option to understand things and make decisions in a human-like manner. A few types of Artificial Neural Networks are Feedforward Neural Network, Modular Neural Network, etc. Applications of Artificial Neural Networks are Social Media, Marketing and Sales, Healthcare etc.

Artificial Neural Network primarily consists of three layers:

1. **Input Layer:** As the name suggests, it accepts inputs in several different formats provided by the programmer.
2. **Hidden Layer:** The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.
3. **Output Layer:** The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

[Reference: https://www.javatpoint.com , https://www.geeksforgeeks.org]



Neural Networks Architecture

[Img source: https://www.geeksforgeeks.org]

Logistic Regression:

Logistic regression is a data analysis technique that uses mathematics to find the relationships between two data factors. It then uses this relationship to predict the value of one of those factors based on the other. The prediction usually has a finite number of outcomes, like yes or no. In simple words, logistic regression is a statistical supervised machine learning algorithm which analyze the relationship between two data factors.

Example: Let's say you want to guess if your website visitor will click the checkout button in their shopping cart or not. Logistic regression analysis looks at past visitor behavior, such as time spent on the website and the number of items in the cart. It determines that, in the past, if visitors spent more than five minutes on the site and added more than three items to the cart, they clicked the checkout button. Using this information, the logistic regression function can then predict the behavior of a new website visitor.

Applications of logistic regression - Manufacturing, Healthcare, Finance, etc.

Types of Logistic Regression: On the basis of the categories, Logistic Regression can be classified into three types:

1. Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

2. Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

3. Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

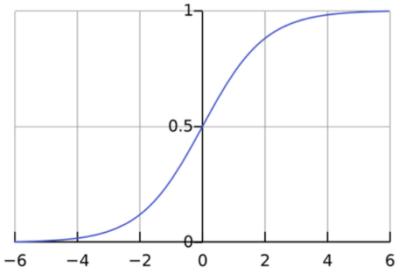
[Reference: https://aws.amazon.com , https://www.geeksforgeeks.org]

Logistic Regression Function: Logistic regression uses a logistic function called a sigmoid function to map predictions and their probabilities. The sigmoid function where the input will be z and we find the probability between 0 and 1. i.e. predicted y. Here, z=w·x+b (z is the weighted sum of inputs. w is the weight vector, x is the input feature vector, and b is the bias term).

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

[Reference: https://www.spiceworks.com]

If we plot this logistic regression equation, we will get an S-curve as shown below-



As we can see, the logit function returns only values between 0 and 1 for the dependent variable, irrespective of the values of the independent variable. This is how logistic regression estimates the value of the dependent variable.

[Reference: https://aws.amazon.com]

Weights and Bias in Neural Networks:

Weights and biases serve as the adjustable parameters in neural networks. They play a central role in determining how the network processes and learns from data. Weights (w) control the strength of connections between neurons and capture relationships between input features and target outputs. Biases (b) introduce adaptability and flexibility, allowing neurons to activate in response to various input conditions.

During training, weights and biases are adjusted through an optimization process, often using a technique called gradient descent. The network calculates the gradient of the error (the difference between its predictions and the true values) with respect to the weights and biases.

[Reference: https://www.geeksforgeeks.org]

Gradient Descent: Machine learning requires the use of a cost function along with gradient descent. As the machine learns to perform a task, the cost function tells the machine how wrong its output is, and gradient descent provides a way for the machine's neural network to adjust the strength of the connections between neurons to improve the machine's accuracy.

Gradient descent is an optimization algorithm used in machine learning to minimize the cost function by iteratively adjusting parameters in the direction of the negative gradient, aiming to find the optimal set of parameters. The cost function represents the discrepancy between the predicted output of the model and the actual output. Gradient descent aims to find the parameters that minimize this discrepancy and improve the model's performance.

[Reference: https://www.linkedin.com , https://www.analyticsvidhya.com]

Mathematically, the update rule for gradient descent can be expressed as:

$$\theta = \theta - \alpha \cdot \nabla J(\theta)$$

Where, θ - the parameters (weights) of the model.

α - the learning rate, determining the size of the steps we take during optimization.

J(θ) - the cost function (loss) we aim to minimize.

∇J(θ) - the gradient of the cost function with respect to the parameters

[Reference: https://medium.com]

Learning Rate: Alpha (α), also known as the learning rate, is a hyperparameter that determines the step size at each iteration of the gradient descent algorithm. It controls how quickly or slowly the algorithm converges to the optimal solution. The value of alpha is crucial as it can greatly impact the optimization process.

[Reference: https://medium.com]

Perceptron:

Perceptron is one of the simplest artificial neural network architectures. It was introduced by Frank Rosenblatt in 1957s. It is the simplest type of feedforward neural network, consisting of a single layer of input nodes that are fully connected to a layer of output nodes. A perceptron is essentially a linear classifier used for binary classification tasks, which

means it predicts whether an input belongs to one class or another based on a linear predictor function combining a set of weights with the feature vector.

Types of Perceptron:

Single-Layer Perceptron: This type of perceptron is limited to learning linearly separable patterns effective for tasks where the data can be divided into distinct categories through a straight line.

Multilayer Perceptron: Multilayer perceptron possess enhanced processing capabilities as they consist of two or more layers, adept at handling more complex patterns and relationships within the data.

The basic structure of a perceptron includes-

1. **Inputs ($x_0, x_1, ..., x_n$):** These are analogous to the dendrites, where x0 is often used to represent the bias input.
2. **Weights ($w_0, w_1, ..., w_n$):** Each input is multiplied by its respective weight, analogous to the synaptic strength in biological neurons. w0 is the weight for the bias.
3. **Bias**: The bias, or threshold allows the perceptron to shift the decision boundary away from the origin without depending on the input values alone.
4. **Summation (∑):** This is the process of summing all the weighted inputs, along with the bias weight.
5. **Activation Function (f):** This function determines the output of the perceptron, based on the weighted sum. If the sum is above a threshold, the perceptron 'fires' (usually outputs 1); otherwise, it 'does not fire' (outputs 0).
6. **Output:** The final binary result after the activation function has been applied.

[Reference: https://www.linkedin.com , https://www.geeksforgeeks.org]

Suppose we are attempting to learn the AND operator for the following input-class pairs $((x_1, x_2), d_i) : ((0,0),0), ((0,1),0), ((1,0),0)$, and $((1,1),1)$. Let us use a learning rate of $\alpha = 0.5$ and run through the algorithm until we can classify all four points correctly.

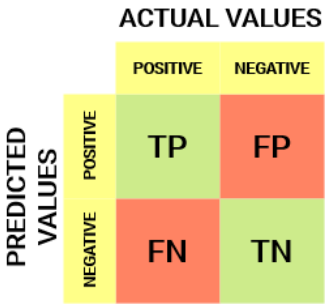
w(0) = [0, 0], b(0) = 0

1	w(0) = [0, 0], b(0) = 0	y = [0, 0, 0, 0]	w(1) = [0.5, 0.5], b(1) = 0.5
2	w(1) = [0.5, 0.5], b(1) = 0.5	y = [1, 1, 1, 1]	w(2) = [0, 0]; b(2) = -1
3	w(2) = [0, 0], b(2) = -1	y = [0, 0, 0, 0]	w(3) = [0.5, 0.5], b(3) = -0.5
4	w(3) = [0.5, 0.5], b(3) = -0.5	y = [0, 0, 0, 1]	SUCCESS!

[Reference: https://brilliant.org]

Confusion matrix:

A confusion matrix is a performance evaluation tool in machine learning, representing the accuracy of a classification model. It displays the number of true positives, true negatives, false positives, and false negatives. A confusion matrix visualizes and summarizes the performance of a classification algorithm.



a 2 x 2 confusion matrix

[Img source: https://www.analyticsvidhya.com]

Important Terms in a Confusion Matrix-

1. True Positive (TP): These are the cases where the model correctly predicts the positive class.
   Example: In a medical test for detecting a disease, TP is the number of sick patients correctly identified as sick.
2. True Negative (TN): These are the cases where the model correctly predicts the negative class.
   Example: TN is the number of healthy patients correctly identified as healthy.
3. False Positive (FP): These are the cases where the model incorrectly predicts the positive class also known as the type I error.
   Example: FP is the number of healthy patients incorrectly identified as sick.
4. False Negative (FN): These are the cases where the model incorrectly predicts the negative class also known as the type II error.
   Example: FN is the number of sick patients incorrectly identified as healthy.

[Reference: https://www.sciencedirect.com , https://www.analyticsvidhya.com]

Accuracy: Accuracy is the most common metric to be used in everyday talk. Accuracy answers the question "Out of all the predictions we made, how many were true?"

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Precision is a metric that gives you the proportion of true positives to the amount of total positives that the model predicts. It answers the question "Out of all the positive predictions we made, how many were true?"

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: Recall focuses on how good the model is at finding all the positives. Recall is also called true positive rate and answers the question "Out of all the data points that should be predicted as true, how many did we correctly predict as true?"

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score: F1 Score is a measure that combines recall and precision. As we have seen there is a trade-off between precision and recall, F1 can therefore be used to measure how effectively our models make that trade-off.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

[Reference: https://sv.labelf.ai]

Naïve Bayes:

Naïve Bayes is part of a family of generative learning algorithms, meaning that it seeks to model the distribution of inputs of a given class or category. Unlike discriminative classifiers, like logistic regression, it does not learn which features are most important to differentiate between classes. Naïve Bayes algorithm is used for classification problems. It is highly used in text classification. Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle.

Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. The formula of this theorem is given below-

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

A, B = events
$P(A|B)$ = probability of A given B is true
$P(B|A)$ = probability of B given A is true
$P(A), P(B)$ = the independent probabilities of A and B

[Reference: https://en.wikipedia.org , https://www.ibm.com]

Types of Naïve Bayes Classifiers:

- Gaussian Naïve Bayes: Assumes that continuous features follow a Gaussian (normal) distribution.
- Multinomial Naïve Bayes: Used for discrete counts, like word counts in text classification.
- Bernoulli Naïve Bayes: Used for binary or boolean features.

Naive Bayes theorem –

$$P(y|x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

This can be expressed mathematically as –

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Applications of Naive Bayes Classifier:

- Spam Email Filtering
- Text Classification
- Medical Diagnosis
- Credit Scoring
- Weather Prediction

[Reference: https://www.geeksforgeeks.org , https://www.ibm.com]