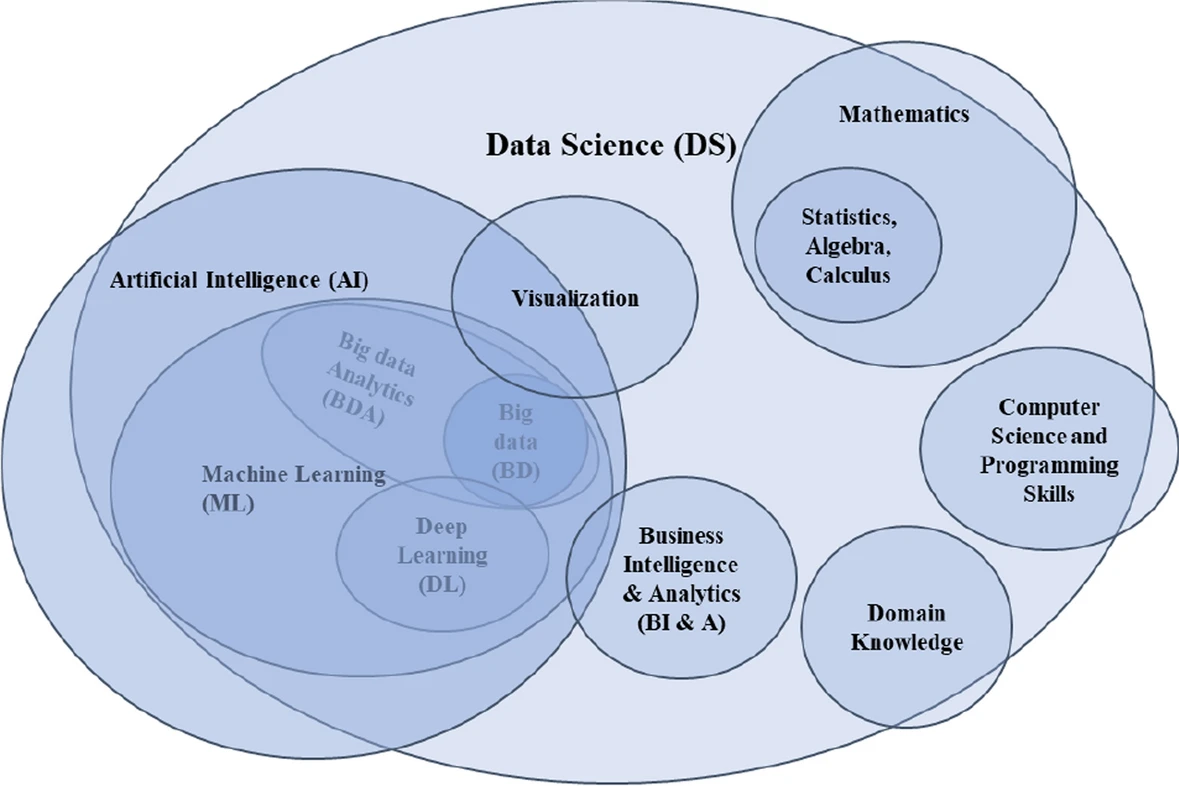
****

# Math

**Linear algebra:**

1. Vector
2. Metrices
3. Operation (addition, multiplication, inversion)
4. Eigenvalues and Eigenvector

**Calculus:**

1. Differentiation and Integration
2. Partial derivatives and gradient descent.

# Statistics

**Probability:**

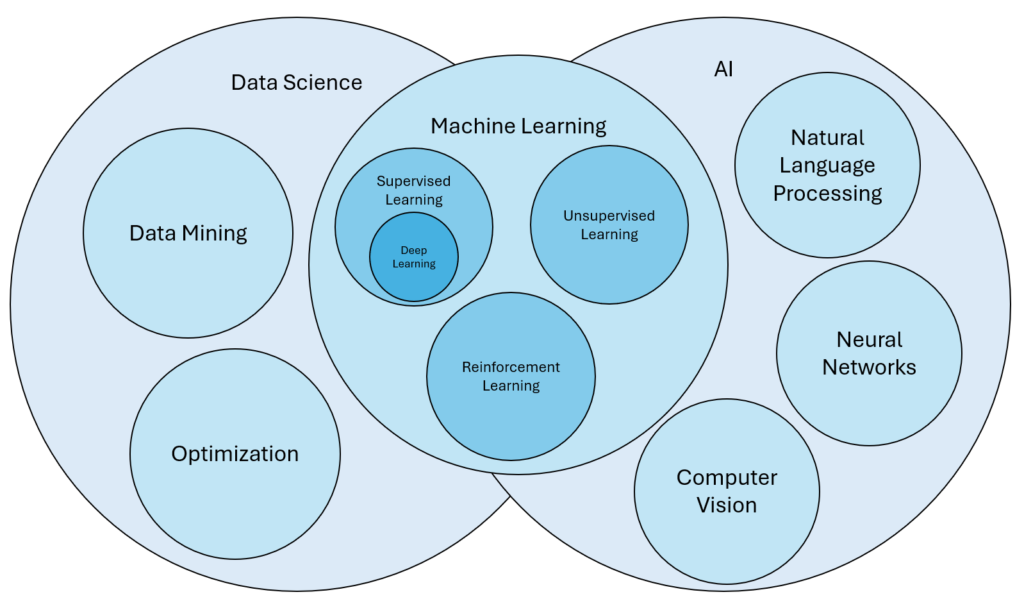
1. Normal
2. Binomial
3. Poisson

**Bayes theorem:**

**Variance:**

**Hypothesis testing:**

# Data Handling and Visualization

****

# Machine Learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed.

**Supervised Learning:**

1. Classification

* Logistic Regression
* Support Vector Machine
* Random Forest
* Decision Tree
* K-Nearest Neighbors (KNN)
* Naive Bayes

1. Regression

* Linear Regression
* Polynomial Regression
* Ridge Regression
* Lasso Regression
* Decision tree
* Random Forest

**Unsupervised Learning:**

1. Clustering:

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

1. Association:

* Apriori Algorithm
* Eclat
* FP-growth Algorithm

**Semi-Supervised Learning:**

* supervised
* unsupervised

**Reinforcement Learning:**

1. Q-Learning,
2. Deep Q-Networks (DQN).

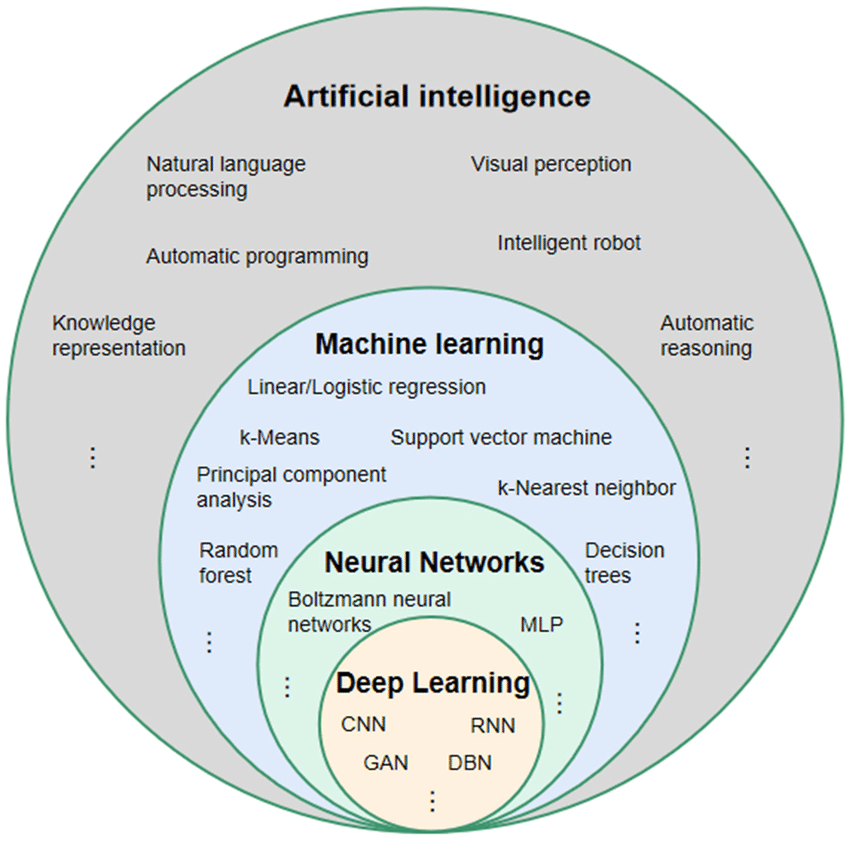
# Advanced Machine Learning

Deep Learning:

1. Neural Networks: Basics of neural network architecture and training.
2. Convolutional Neural Networks (CNNs): For image recognition tasks.
3. Recurrent Neural Networks (RNNs): For sequential data.

Natural Language Processing (NLP):

1. Text preprocessing: tokenization, stemming, lemmatization.
2. Techniques: Bag of Words, TF-IDF, Word Embeddings (Word2Vec, GloVe).
3. Applications: sentiment analysis, text classification.



# Supervised Learning:

Supervised Machine Learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output Y = f(X). The goal is to approximate the mapping function so well that when you have new input data (x) you can predict the output variables (Y) for that data. Supervised learning is a fundamental approach in machine learning where models are trained on labeled datasets.

1. Classification:
2. Regression:

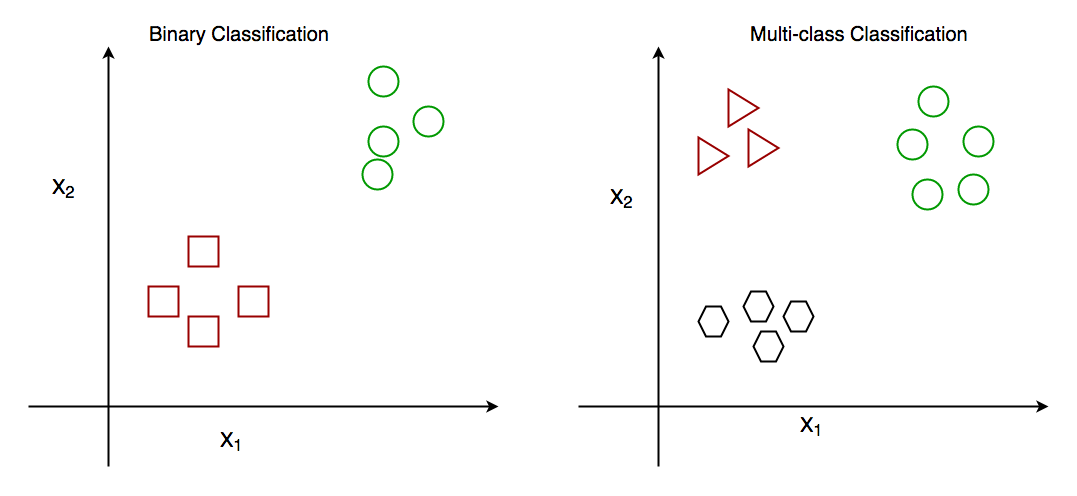
## Classification

Machine Learning classification is a type of supervised learning technique where an algorithm is trained on a labeled dataset to predict the class or category of new, unseen data.

The main objective of classification machine learning is to build a model that can accurately assign a label or category to a new observation based on its features.

Classification Types:

1. Binary Classification: binary classification, the goal is to classify the input into one of two classes or categories. Example – On the basis of the given health conditions of a person, we have to determine whether the person has a certain disease or not.
2. Multiclass Classification: multi-class classification, the goal is to classify the input into one of several classes or categories. For Example – On the basis of data about different species of flowers, we have to determine which specie our observation belongs to.



1. Multi-Label Classification: Multi-label Classification the goal is to predict which of several labels a new data point belongs to. This is different from multiclass classification, where each data point can only belong to one class. For example, a multi-label classification algorithm could be used to classify images of animals as belonging to one or more of the categories cat, dog, bird, or fish.
2. Imbalanced Classification: Imbalanced Classification the goal is to predict whether a new data point belongs to a minority class, even though there are many more examples of the majority class. For example, a medical diagnosis algorithm could be used to predict whether a patient has a rare disease, even though there are many more patients with common diseases.

### Classification **Algorithms**

There are various types of classifiers algorithms. Some of them are:

1. Linear Classifiers: Linear models create a linear decision boundary between classes. They are simple and computationally efficient. Some of the linear classification models are as follows:
2. Logistic Regression
3. Support Vector Machines having kernel = ‘linear’
4. Single-layer Perceptron
5. Stochastic Gradient Descent (SGD) Classifier
6. Non-linear Classifiers: Non-linear models create a non-linear decision boundary between classes. They can capture more complex relationships between the input features and the target variable. Some of the non-linear classification models are as follows:
7. K-Nearest Neighbours
8. Kernel SVM
9. Naive Bayes
10. Decision Tree Classification
11. Ensemble learning classifiers:
12. Random Forests,
13. AdaBoost,
14. Bagging Classifier,
15. Voting Classifier,
16. ExtraTrees Classifier
17. Multi-layer Artificial Neural Networks

In machine learning, classification learners can also be classified as either “lazy” or “eager” learners.

1. Lazy Learners: Lazy Learners are also known as instance-based learners, lazy learners do not learn a model during the training phase. Instead, they simply store the training data and use it to classify new instances at prediction time. It is very fast at prediction time because it does not require computations during the predictions. it is less effective in high-dimensional spaces or when the number of training instances is large. Examples of lazy learners include k-nearest neighbors and case-based reasoning.
2. Eager Learners: Eager Learners are also known as model-based learners, eager learners learn a model from the training data during the training phase and use this model to classify new instances at prediction time. It is more effective in high-dimensional spaces having large training datasets. Examples of eager learners include decision trees, random forests, and support vector machines.

### Classification Models

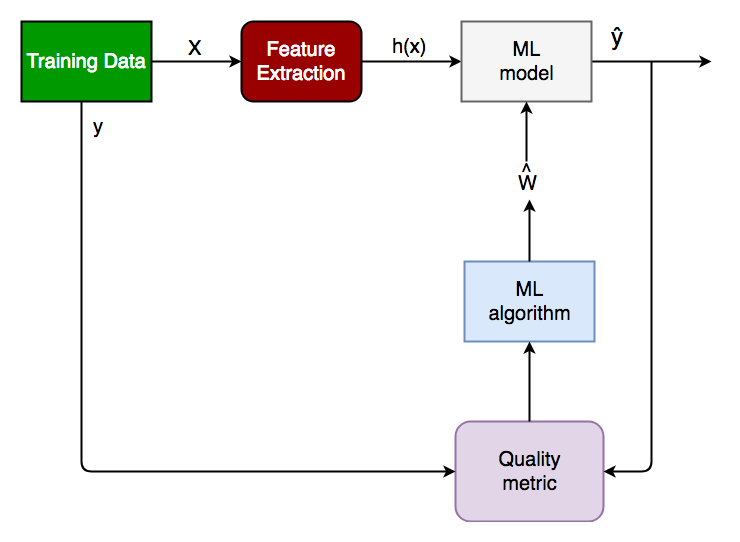
1. Classification Accuracy: The proportion of correctly classified instances over the total number of instances in the test set. It is a simple and intuitive metric but can be misleading in imbalanced datasets where the majority class dominates the accuracy score.
2. Confusion matrix: A table that shows the number of true positives, true negatives, false positives, and false negatives for each class, which can be used to calculate various evaluation metrics.
3. Precision and Recall: Precision measures the proportion of true positives over the total number of predicted positives, while recall measures the proportion of true positives over the total number of actual positives. These metrics are useful in scenarios where one class is more important than the other, or when there is a trade-off between false positives and false negatives.
4. F1-Score: The harmonic means of precision and recall, calculated as 2 x (precision x recall) / (precision + recall). It is a useful metric for imbalanced datasets where both precision and recall are important.
5. ROC curve and AUC: The Receiver Operating Characteristic (ROC) curve is a plot of the true positive rate (recall) against the false positive rate (1-specificity) for different threshold values of the classifier’s decision function. The Area Under the Curve (AUC) measures the overall performance of the classifier, with values ranging from 0.5 (random guessing) to 1 (perfect classification).
6. Cross-validation: A technique that divides the data into multiple folds and trains the model on each fold while testing on the others, to obtain a more robust estimate of the model’s performance.

### Characteristics of Classification

1. Categorical Target Variable: Classification deals with predicting categorical target variables that represent discrete classes or labels. Examples include classifying emails as spam or not spam, predicting whether a patient has a high risk of heart disease, or identifying image objects.
2. Accuracy and Error Rates: Classification models are evaluated based on their ability to correctly classify data points. Common metrics include accuracy, precision, recall, and F1-score.
3. Model Complexity: Classification models range from simple linear classifiers to more complex nonlinear models. The choice of model complexity depends on the complexity of the relationship between the input features and the target variable.
4. Overfitting and Underfitting: Classification models are susceptible to overfitting and underfitting. Overfitting occurs when the model learns the training data too well and fails to generalize to new data.

### How does Classification Machine Learning Work?

The basic idea behind classification is to train a model on a labeled dataset, where the input data is associated with their corresponding output labels, to learn the patterns and relationships between the input data and output labels. Once the model is trained, it can be used to predict the output labels for new unseen data.



### **Implementation of Classification Model in Machine Learning**

1. **Problem:** Before getting started with classification, it is important to understand the problem you are trying to solve. What are the class labels you are trying to predict? What is the relationship between the input data and the class labels?

Suppose we have to predict whether a patient has a certain disease or not, on the basis of 7 independent variables, called features. This means, there can be only two possible outcomes:

The patient has the disease, which means “True”.

The patient has no disease. which means “False”.

1. **Data preparation:** Once you have a good understanding of the problem, the next step is to prepare your data. This includes collecting and preprocessing the data and splitting it into training, validation, and test sets. In this step, the data is cleaned, preprocessed, and transformed into a format that can be used by the classification algorithm.

X: It is the independent feature, in the form of an N\*M matrix. N is the no. of observations and M is the number of features.

y: An N vector corresponding to predicted classes for each of the N observations.

1. **Feature Extraction:** The relevant features or attributes are extracted from the data that can be used to differentiate between the different classes.

Suppose our input X has 7 independent features, having only 5 features influencing the label or target values remaining 2 are negligibly or not correlated, then we will use only these 5 features only for the model training.

1. **Model Selection:** There are many different models that can be used for classification, including logistic regression, decision trees, support vector machines (SVM), or neural networks. It is important to select a model that is appropriate for your problem, taking into account the size and complexity of your data, and the computational resources you have available.
2. **Model Training:** Once you have selected a model, the next step is to train it on your training data. This involves adjusting the parameters of the model to minimize the error between the predicted class labels and the actual class labels for the training data.
3. **Model Evaluation:** Evaluating the model: After training the model, it is important to evaluate its performance on a validation set. This will give you a good idea of how well the model is likely to perform on new, unseen data.

Log Loss or Cross-Entropy Loss, Confusion Matrix, Precision, Recall, and AUC-ROC curve are the quality metrics used for measuring the performance of the model.

1. **Fine-tuning the model:** If the model’s performance is not satisfactory, you can fine-tune it by adjusting the parameters, or trying a different model.
2. **Deploying the model:** Finally, once we are satisfied with the performance of the model, we can deploy it to make predictions on new data.  it can be used for real world problem.
3. **Examples of Machine Learning Classification in Real Life**
4. Email spam filtering
5. Credit risk assessment
6. Medical diagnosis
7. Image classification
8. Sentiment analysis.
9. Fraud detection
10. Quality control
11. Recommendation systems

Code:

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

from sklearn import datasets

from sklearn import svm

from sklearn.tree import DecisionTreeClassifier

from sklearn.naive\_bayes import GaussianNB

# import the iris dataset

iris = datasets.load\_iris()

X = iris.data

y = iris.target

# splitting X and y into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    X, y, test\_size=0.3, random\_state=1)

# GAUSSIAN NAIVE BAYES

gnb = GaussianNB()

# train the model

gnb.fit(X\_train, y\_train)

# make predictions

gnb\_pred = gnb.predict(X\_test)

# print the accuracy

print("Accuracy of Gaussian Naive Bayes: ",

      accuracy\_score(y\_test, gnb\_pred))

# print other performance metrics

print("Precision of Gaussian Naive Bayes: ",

      precision\_score(y\_test, gnb\_pred, average='weighted'))

print("Recall of Gaussian Naive Bayes: ",

      recall\_score(y\_test, gnb\_pred, average='weighted'))

print("F1-Score of Gaussian Naive Bayes: ",

      f1\_score(y\_test, gnb\_pred, average='weighted'))

# DECISION TREE CLASSIFIER

dt = DecisionTreeClassifier(random\_state=0)

# train the model

dt.fit(X\_train, y\_train)

# make predictions

dt\_pred = dt.predict(X\_test)

# print the accuracy

print("Accuracy of Decision Tree Classifier: ",

      accuracy\_score(y\_test, dt\_pred))

# print other performance metrics

print("Precision of Decision Tree Classifier: ",

      precision\_score(y\_test, dt\_pred, average='weighted'))

print("Recall of Decision Tree Classifier: ",

      recall\_score(y\_test, dt\_pred, average='weighted'))

print("F1-Score of Decision Tree Classifier: ",

      f1\_score(y\_test, dt\_pred, average='weighted'))

# SUPPORT VECTOR MACHINE

svm\_clf = svm.SVC(kernel='linear')  # Linear Kernel

# train the model

svm\_clf.fit(X\_train, y\_train)

# make predictions

svm\_clf\_pred = svm\_clf.predict(X\_test)

# print the accuracy

print("Accuracy of Support Vector Machine: ",

       accuracy\_score(y\_test, svm\_clf\_pred))

# print other performance metrics

print("Precision of Support Vector Machine: ",

       precision\_score(y\_test, svm\_clf\_pred, average='weighted'))

print("Recall of Support Vector Machine: ",

       recall\_score(y\_test, svm\_clf\_pred, average='weighted'))

print("F1-Score of Support Vector Machine: ",

       f1\_score(y\_test, svm\_clf\_pred, average='weighted'))

## Regression

Regression is a statistical approach used to analyze the relationship between a dependent variable (target variable) and one or more independent variables (predictor variables). The objective is to determine the most suitable function that characterizes the connection between these variables.