Programme Name: BCA Semester V
Course Name & Code: Machine Learning & BCA57204
Class: BCA2023
Academic Session: 2025-26

Study Material

Module III: Training of Model in Machine Learning [PART-1]

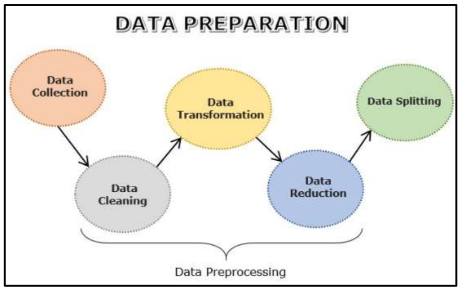
Training in Machine Learning

Machine learning is a branch of artificial intelligence that allows computers to learn patterns and make predictions without being explicitly programmed. The central idea is that a model can be trained using data, allowing it to improve its performance over time. In this module, we will explore how models are trained, what role algorithms play, the different types of algorithms, how to prepare and select data, and the steps involved in the machine learning cycle.

At the core of machine learning is the model, which is essentially a mathematical function. The model receives input data, processes it, and produces an output. The accuracy of this output depends on how well the model has been trained. Training means adjusting the internal parameters of the model so that its predictions match real outcomes as closely as possible. For example, consider an email spam filter. The input data is the text of emails, and the output is whether an email is “spam” or “not spam.” The model is trained on a large dataset of emails that are already labeled. By analyzing patterns in the data, the model learns to make predictions on new, unseen emails.

Data Preparation

A key part of machine learning is preparing the data properly. Raw data may or may not contain errors and inconsistencies. Hence, drawing actionable insights is not straightforward. We have to prepare the data to rescue us from the pitfalls of incomplete, inaccurate, and unstructured data.Data preparation is the process of making raw data ready for after processing and analysis. The key methods are to collect, clean, and label raw data in a format suitable for machine learning (ML) algorithms, followed by data exploration and visualization. For example, categorical values may be converted into numerical form, and numerical features may be standardized so that they are on the same scale.

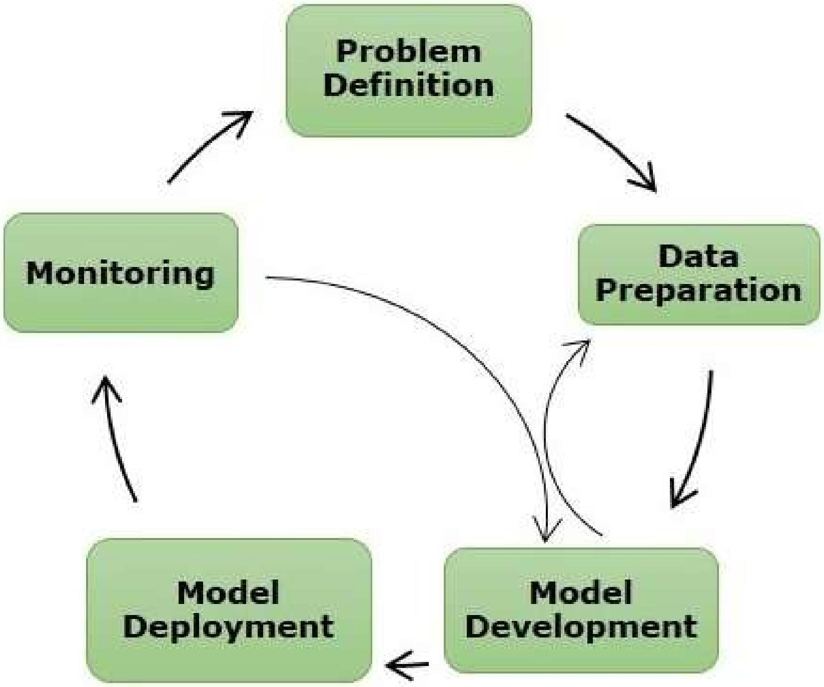


The process of cleaning and combining raw data before using it for machine learning and business analysis is known as data preparation, or sometimes "pre-processing." But it may not be the most attractive of duties, careful data preparation is essential to the success of data analytics. Clear and important ideas from raw data require careful validation, cleaning, and an addition. Any business analysis or model created will only be as strong and validating as the very first information preparation.

Identifying Relevant Data

Not all data is equally useful. Choosing the right data is critical because irrelevant or low-quality data can mislead the model. Feature selection techniques are often applied to identify the most important attributes. For instance, in predicting house prices, features like location and size are far more relevant than the color of the walls.

Machine Learning Cycle



The machine learning life cycle is an iterative process that moves from a business problem to a machine learning solution. It is used as a guide for developing a machine learning project to solve a problem. It provides us with instructions and best practices to be used in each phase while developing ML solutions.

The machine learning life cycle is a process that involves several phases from problem identification to model deployment and monitoring. While developing an ML project, each step in the life cycle is revisited many times through these phases. The stages/ phases involved in the end-to-end machine life cycle process are as follows –

- Problem Definition
- Data Preparation
- Model Development
- Model Deployment
- Monitoring and Maintenance

Machine Learning Algorithms

Machine Learning algorithms are programs that can learn the hidden patterns from the data, predict the output, and improve their performance from experience on their own. Different algorithms can be used in machine learning for different tasks, such as simple linear regression, which can be used for prediction problems like stock marketprediction, and the KNN algorithm can be used for classification problems.

Role of Machine Learning Algorithms

Algorithms are at the heart of machine learning. They provide the instructions or methods that the model uses to learn from data. Different algorithms are suited to different tasks. The choice of algorithm depends on the problem, the size and type of data, and the accuracy required.
Some of the key roles are -

- **Pattern Recognition and Insight Generation:** Algorithms analyze complex data to find hidden patterns, trends, and correlations that humans might miss, providing valuable insights into data behavior.
- **Predictive Modeling:** They learn from past data to predict future outcomes, such as forecasting demand, predicting customer behavior, or identifying potential equipment failures.
- **Classification and Categorization:** Algorithms can sort data into predefined categories, a process used for tasks like spam filtering, image recognition, and disease diagnosis.
- **Decision Automation:** By analyzing data and making predictions, ML algorithms automate complex decision-making processes, leading to more efficient operations in fields like logistics, energy management, and human resources.
- **Personalization and Recommendations:** They power personalized user experiences by suggesting relevant products, movies, or content based on user preferences and past behavior, as seen in e-commerce and streaming platforms.

List of Popular Machine Learning Algorithms

Machine learning algorithms are broadly categorized into three types:

1. Supervised Learning: Algorithms learn from labeled data, where the input-output relationship is known.
2. Unsupervised Learning: Algorithms work with unlabeled data to identify patterns or groupings.
3. Reinforcement Learning: Algorithms learn by interacting with an environment and receiving feedback in the form of rewards or penalties.

1. Supervised Learning:

(i) Linear Regression - Used for predicting a continuous target variable based on one or more independent variables, assuming a linear relationship.

(ii) Logistic Regression - Employed for binary classification tasks, predicting the probability of an event occurring.

(iii) **K-Nearest Neighbors (KNN)** - A non-parametric, instance-based learning algorithm used for both classification and regression, which classifies a data point based on the majority class of its 'k' nearest neighbors.

(iv) **Decision Tree** - A non-parametric supervised learning method used for both classification and regression, which constructs a tree-like model of decisions and their possible consequences.

(v) **Random Forest** - An ensemble learning method that builds multiple decision trees during training and outputs the mode of the classes (for classification) or mean prediction (for regression) of the individual trees.

(vi) **Support Vector Machine (SVM)** - A powerful algorithm for classification and regression tasks, particularly effective in high-dimensional spaces, by finding the optimal hyperplane that separates data points.

(vii) **Naïve Bayes** - A probabilistic classification algorithm based on Bayes' theorem with the "naïve" assumption of independence between features, often used in text classification.

2. Unsupervised Learning:

(i) **K-Means Clustering** - An unsupervised clustering algorithm that partitions 'n' observations into 'k' clusters, where each observation belongs to the cluster with the nearest mean.

(ii) **Principal Component Analysis (PCA)** - A dimensionality reduction technique that transforms data into a new set of orthogonal variables called principal components, capturing the most variance in the data.

3. Neural Networks:

(i) **Backpropagation:** The core algorithm used to train artificial neural networks, adjusting the weights of connections in the network by propagating errors backward through the layers.

Regression Model

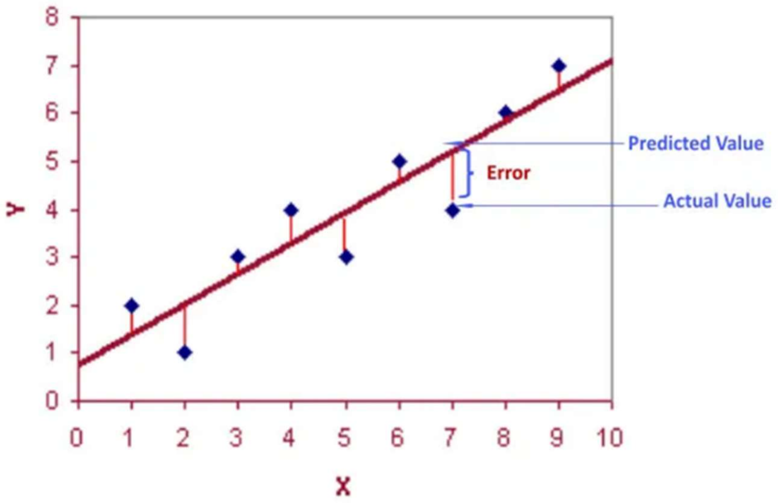
Regression in machine learning refers to a supervised learning technique where the goal is to predict a continuous numerical value based on one or more independent features. It finds relationships between variables so that predictions can be made. we have two types of variables present in regression:

- Dependent Variable (Target): The variable we are trying to predict, e.g house price.
- Independent Variables (Features): The input variables that influence the prediction, e.g locality, number of rooms.

Regression analysis problem works with if the output variable is a real or continuous value, such as “salary” or “weight”. Many different regression models can be used but the simplest model among them is linear regression.

Evaluation Metrics for Regression Model

The performance of a regression model can be understood by knowing the error rate of the predictions made by the model. We can measure the performance by knowing how well the regression line fit the dataset. A good regression model is one where the difference between the actual or observed values and predicted values for the selected model is small and unbiased for train, validation and test data sets.



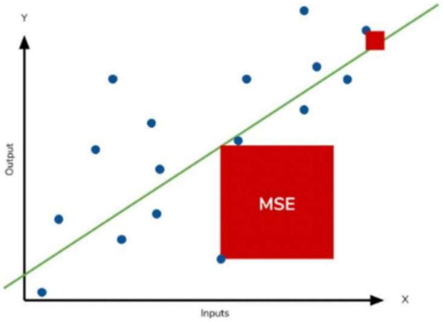
Error = actual value – predicted value

$$e = y - \hat{y}$$

A variety of evaluation measures can be used to determine the strength of any linear regression model. These assessment metrics often give an indication of how well the model is producing the observed outputs.

The most common measurements are:

1. Mean Square Error (MSE) – This is the average of the squared differences between the predicted and actual values. The difference is squared to ensure that both negative and positive differences are accounted for. It gives more weight to larger differences. Particularly useful when we encounter unexpected values that we want to account for.



In mathematical notation, it can be expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$$

Here,

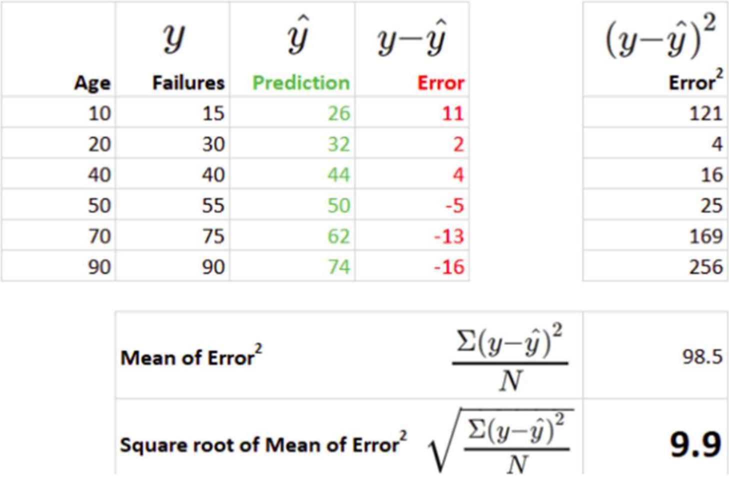
- n is the number of data points.
- y_i is the actual or observed value for the i^{th} data point.
- $\widehat{y_i}$ is the predicted value for the i^{th} data point.

2. Root Mean Squared Error (RMSE) - This is the square root of the MSE. Unlike MSE, it treats all differences equally and is less sensitive to outliers. It describes how well the observed data points match the expected values or the model's absolute fit to the data.

In mathematical notation, it can be expressed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

Example:



	y	$\hat{y}$	$y-\hat{y}$	$(y-\hat{y})^2$
Age	Failures	Prediction	Error	Error^2
10	15	26	11	121
20	30	32	2	4
40	40	44	4	16
50	55	50	-5	25
70	75	62	-13	169
90	90	74	-16	256

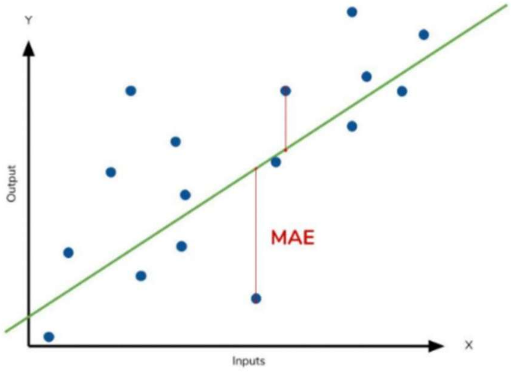
Mean of Error^2	$\frac{\Sigma(y-\hat{y})^2}{N}$	98.5
Square root of Mean of Error^2	$\sqrt{\frac{\Sigma(y-\hat{y})^2}{N}}$	**9.9**

3. Mean Absolute Error (MAE) – This is the simplest of all the metrics. It is an evaluation metric used to calculate the accuracy of a regression model. MAE measures the average absolute difference between the predicted values and actual values.

Mathematically,

$$MAE = \frac{1}{n}\sum_{i=0}^{n} |Actual_{\mathrm{i}} - Predicted_{\mathrm{i}}|$$

A small MAE suggests the model is great at prediction. A large MAE suggests that the model may have trouble in certain areas. A MAE of 0 means that the model is a perfect predictor of the outputs.



4. R-Squared (R^2) Error – This is a statistical metric frequently used to assess the goodness of fit of a regression model, also known as the Coefficient of Determination. This metric measures the proportion of the total variation in the dependent variable that is captured by the model. In simpler terms, R-Squared tells us the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

Mathematically,

$$R^2 = 1 - \left(\frac{RSS}{TSS}\right)$$

Here,

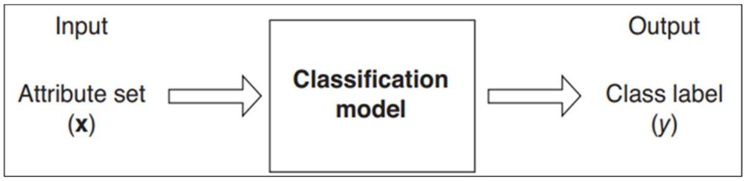
- **Residual sum of Squares (RSS):** The sum of squares of the residual for each data point in the plot or data. It is a measurement of the difference between the output that was observed and what was anticipated.

- **Total Sum of Squares (TSS):** The sum of the differences between the actual values of the dependent variable and the mean value of the dependent variable, all squared.

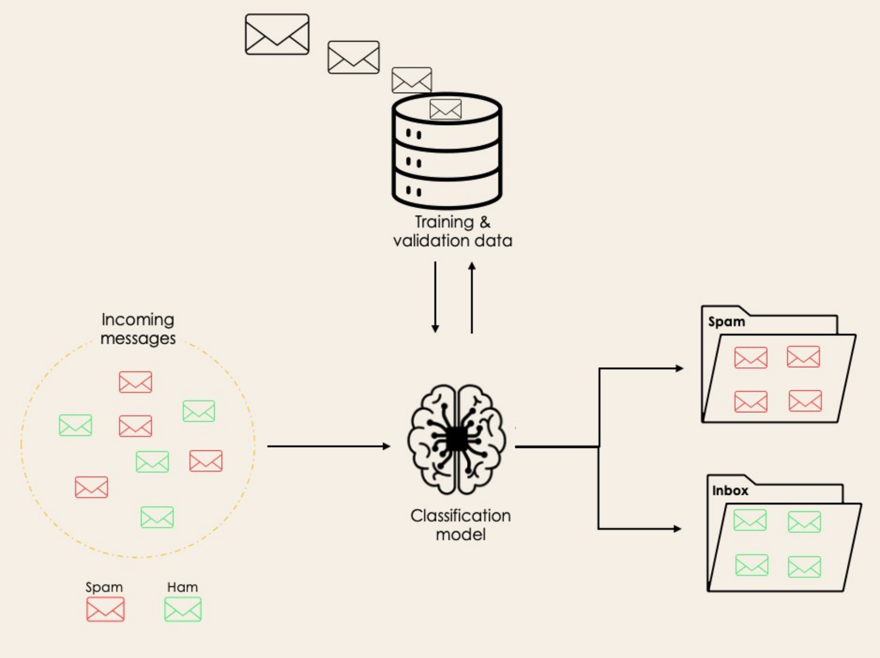
R-squared values range from 0 to 1 (or 0% to 100%). It is a decimal value between 0 and 1, and is often represented as a percentage when discussing model fit. An R-Squared of 100% (or 1) indicates the model perfectly fits the data. An R-Squared of 0% indicates that the dependent variable cannot be predicted from the independent variable(s) at all.

Classification Model

Classification is the task of learning a target function (f) that maps each attribute set x to one of the predefined class labels y. The main purpose is to determine which category or class a new data set belongs to by training a model using labeled data.



Classification teaches a machine to sort things into categories. It learns by looking at examples with labels (like emails marked "spam" or "not spam"). After learning, it can decide which category new items belong to, like identifying if a new email is spam or not. For example a classification model might be trained on a dataset of images labeled as either dogs or cats , and it can be used to predict the class of new and unseen images as dogs or cats based on their features such as color, texture and shape.



The general two-step data classification process includes :

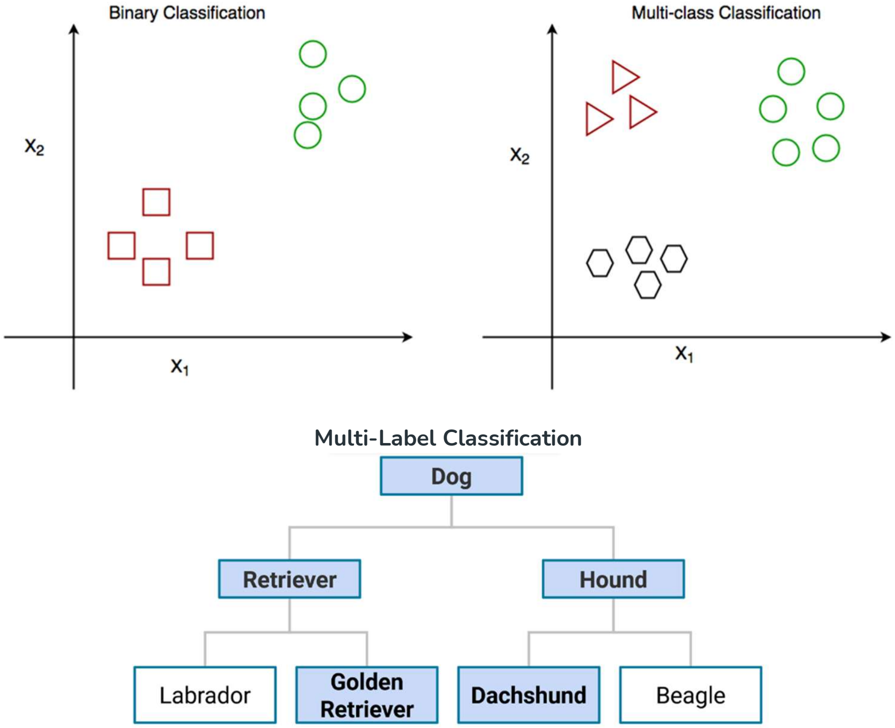
(a) Learning - describing a set of predetermined classes. Each sample is assumed to belong to a predefined class, as determined by the class label attribute. The set of samples used for model construction is the training set.

(b) Classification - for classifying future or unknown objects. The known label of the test sample is compared with the classified result from the model. It estimates the accuracy of the model. If the accuracy is acceptable, use the model to classify new data.

Types of Classification

Classification based predictive modeling tasks are distinguished from each other based on the number of categories and the degree to which the categories are exclusive. There are different types of classification problems depending on how many categories (or classes) we are working with and how they are organized -

1. Binary classification sorts data into two exclusive categories.
2. Multiclass classification sorts data into more than two exclusive categories.
3. Multilabel classification sorts data into nonexclusive categories.



Types of Classification Model

Working of Classification in Machine Learning

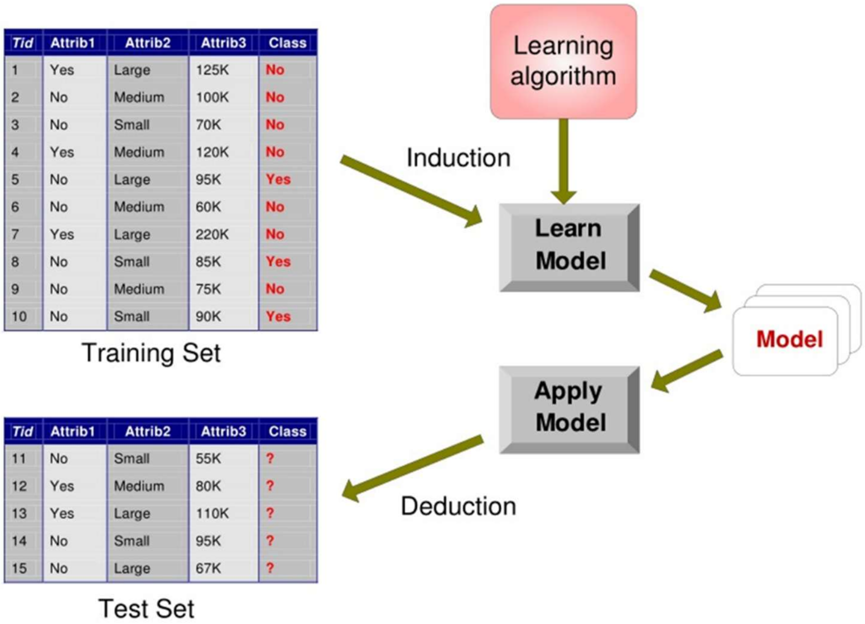
Classification involves training a model using a labeled dataset, where each input is paired with its correct output label. The model learns patterns and relationships in the data, so it can later predict labels for new, unseen inputs. In machine learning, classification works by training a model to learn patterns from labeled data, so it can predict the category or class of new, unseen data.

Here's how it works:

1. **Data Collection**: You start with a dataset where each item is labeled with the correct class (for example, "cat" or "dog").
2. **Feature Extraction**: The system identifies features (like color, shape, or texture) that help

distinguish one class from another. These features are what the model uses to make predictions.

3. **Model Training**: Classification - machine learning algorithm uses the labeled data to learn how to map the features to the correct class. It looks for patterns and relationships in the data.
4. **Model Evaluation**: Once the model is trained, it's tested on new, unseen data to check how accurately it can classify the items.
5. **Prediction**: After being trained and evaluated, the model can be used to predict the class of new data based on the features it has learned.
6. **Model Evaluation**: Evaluating a classification model is a key step in machine learning. It helps us check how well the model performs and how good it is at handling new, unseen data. Depending on the problem and needs we can use different metrics to measure its performance.



Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

General approach to solving a classification problem

Evaluation Metrics for Classification Model

When building machine learning models, it’s important to understand how well they perform. Evaluation metrics help us to measure the effectiveness of our models. Whether we are solving a classification problem, predicting continuous values or clustering data, selecting the right evaluation metric allows us to assess how well the model meets our goals.

- **Accuracy:** provides the proportion of correctly classified instances.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total}$$

- **Precision:** Precision focuses on the accuracy of positive predictions.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

- **Recall (Sensitivity or True Positive Rate):** Recall measures the proportion of correctly predicted positive instances among all actual positive instances.

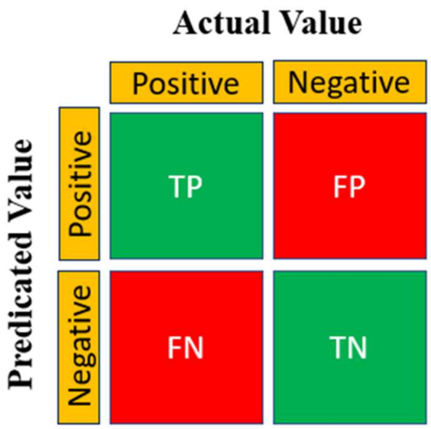
$$Recall = \frac{True\,Positives}{True\,Positives + False\,Negatives}$$

- **F1 Score:** F1 score is the harmonic mean of precision and recall.

$$F1\,Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Confusion Matrix

It is a table used to evaluate the performance of a machine learning algorithm for classification tasks. It is a square matrix that compares the actual and predicted values of a classifier. By examining the values in the confusion matrix, the performance metrics are calculated to evaluate the model's performance. The name "confusion matrix" comes from the fact that it makes it easy to see which classes the model is consistently misclassifying, or "confusing" with other classes.



		Actual Value	
		Positive	Negative
Predicated Value	Positive	TP	FP
	Negative	FN	TN

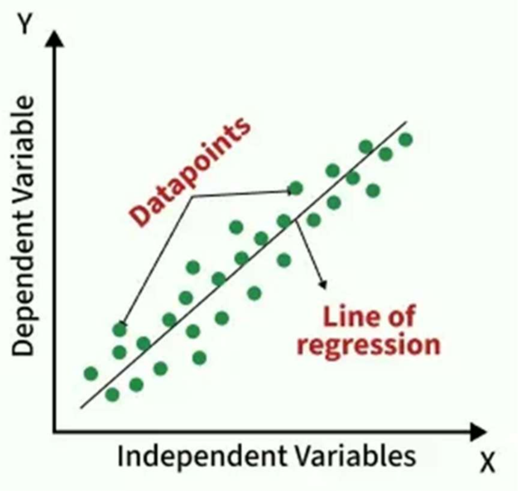
- **True Positive (TP):** It indicates that the predicted value or class is the same as the actual value. The model predicted that the value would be positive.
- **True Negative (TN):** It indicates that the predicted value or class is the same as the actual class. The model predicted that the value would be negative.
- **False Positive (FP) – Type I Error:** The value predicted was incorrect. The model predicted that the value would be positive, even though it was negative. It is also called the Type I error.
- **False Negative (FN) – Type II Error:** The value predicted was incorrect. The model predicted that the value would be negative, even though it was positive. It is also called the Type II error.

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

Confusion Matrix for a 2-class problem

Linear Regression

Linear regression is one of the most popular machine learning algorithms that is used for predictive analysis. This algorithm learns from the labelled datasets and maps the data points with the most optimized linear functions, which can be used for prediction on new datasets. It assumes that there is a linear relationship between the input and output, meaning the output changes at a constant rate as the input changes. This relationship is represented by a straight line.



Best Fit Line in Linear Regression

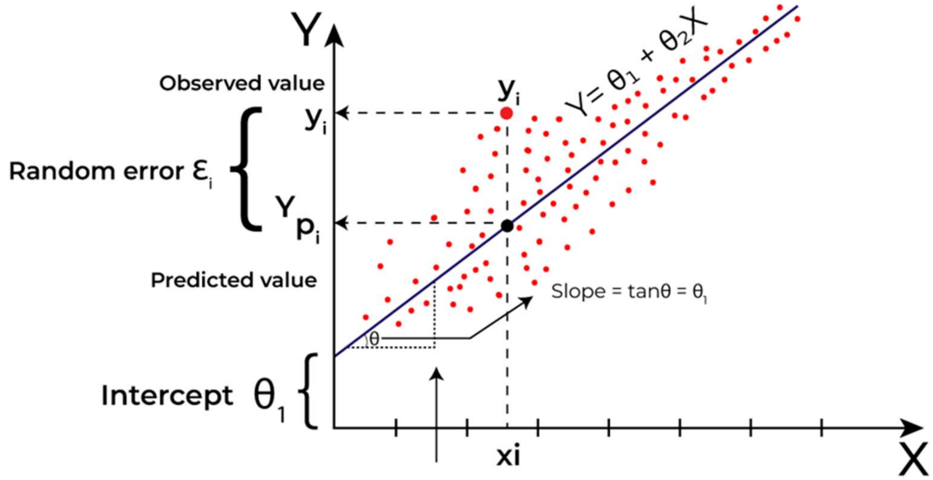
The best-fit line is the straight line that most accurately represents the relationship between the independent variable (input) and the dependent variable (output). It is the line that minimizes the difference between the actual data points and the predicted values from the model. The goal of linear regression is to find a straight line that minimizes the error (the difference) between the observed data points and the predicted values. This line helps us predict the dependent variable for new, unseen data.

For simple linear regression (with one independent variable), the best-fit line is represented by the equation -

$$y = mx + b$$

where,

- **y** is the predicted value (dependent variable)
- **x** is the input (independent variable)
- **m** is the slope of the line (how much y changes when x changes)
- **b** is the intercept (the value of y when x = 0)


Y
Observed value
y_i
Random error ε_i
Y_{p_i}
Predicted value
Intercept θ_1
$Y = \theta_1 + \theta_2 X$
Slope = $\tan\theta = \theta_1$
θ
xi
X

- **Slope (m):** The slope of the best-fit line indicates how much the dependent variable (y) changes with each unit change in the independent variable (x). For example if the slope is 5, it means that for every 1-unit increase in x, the value of y increases by 5 units.

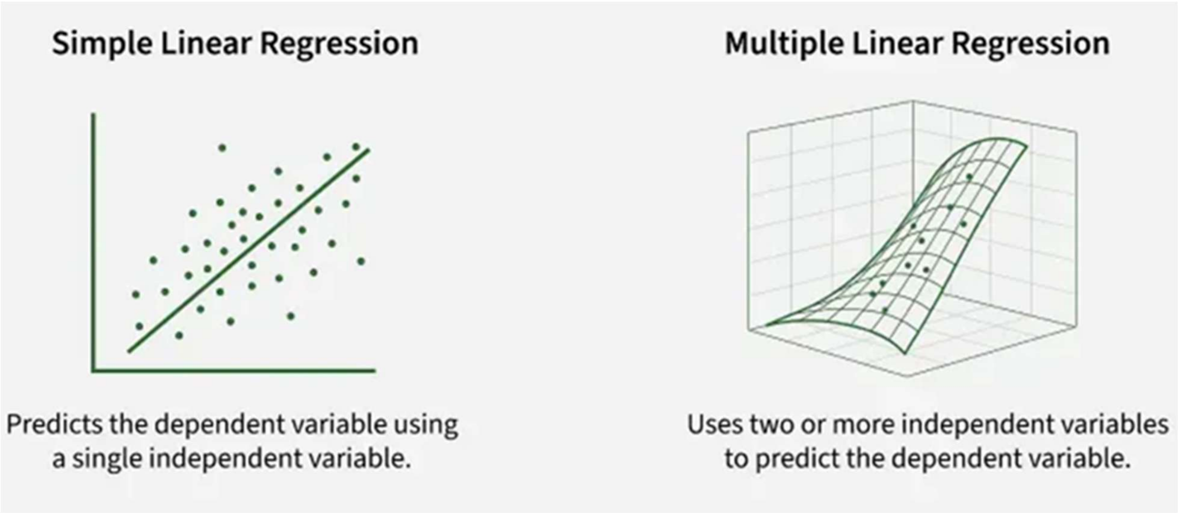
- **Intercept (b):** The intercept represents the predicted value of y when x = 0. It's the point where the line crosses the y-axis.

Types of Linear Regression

Linear regression is further divided into two types:

- **Simple Linear Regression:** In simple linear regression, a single independentvariable is used to predict the value of the dependent variable.

- **Multiple Linear Regression:** In multiple linear regression, more than one independent variable are used to predict the value of the dependent variable.


Simple Linear Regression
Multiple Linear Regression
Predicts the dependent variable using a single independent variable.
Uses two or more independent variables to predict the dependent variable.

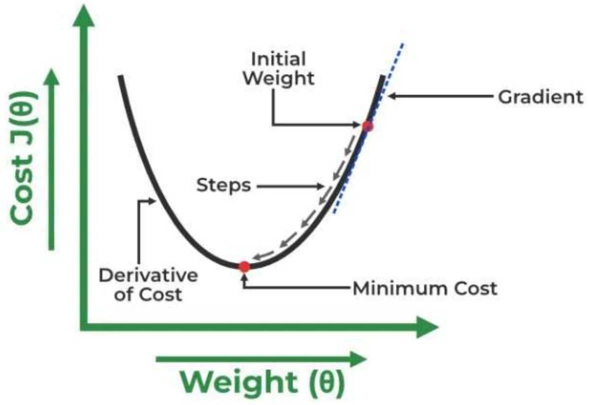
Importance of Linear Regression

Linear regression is important because:

- **Simplicity and Interpretability:** It's easy to understand and interpret, making it a starting point for learning about machine learning.
- **Predictive Ability**: Helps predict future outcomes based on past data, making it useful in various fields like finance, healthcare and marketing.
- **Basis for Other Models:** Many advanced algorithms, like logistic regression or neural networks, build on the concepts of linear regression.
- **Efficiency:** It's computationally efficient and works well for problems with a linear relationship.
- **Widely Used:** It's one of the most widely used techniques in both statistics and machine learning for regression tasks.
- **Analysis:** It provides insights into relationships between variables (e.g., how much one variable influences another).

Gradient Descent for Linear Regression

Gradient descent is an optimization technique used to train a linear regression model by minimizing the prediction error. It works by starting with random model parameters and repeatedly adjusting them to reduce the difference between predicted and actual values.



It works:

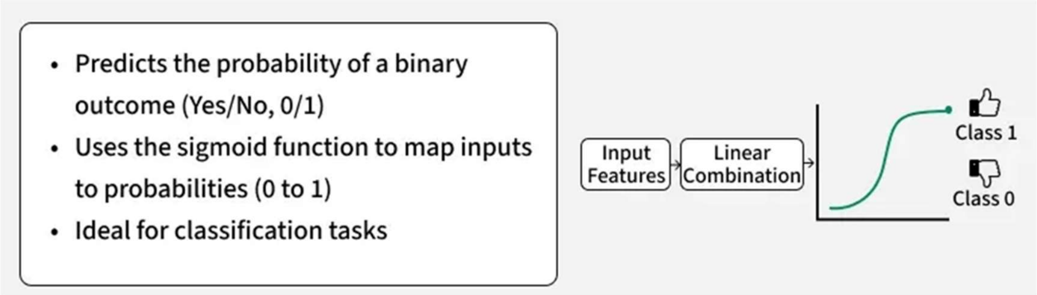
- Start with random values for slope and intercept.
- Calculate the error between predicted and actual values.
- Find how much each parameter contributes to the error (gradient).
- Update the parameters in the direction that reduces the error.
- Repeat until the error is as small as possible.

This helps the model find the best-fit line for the data.

Logistic Regression

Logistic regression is a supervised learning algorithm that is used to predict the categorical variables or discrete values. It can be used for the classification problems in machine learning, and the output of the logistic regression algorithm canbe either Yes or No, 0 or 1, Red or Blue, etc.

It's called 'logistic' because the transformation of the input variables is done using a mathematical function called the logistic function, which creates an S-shaped curve.

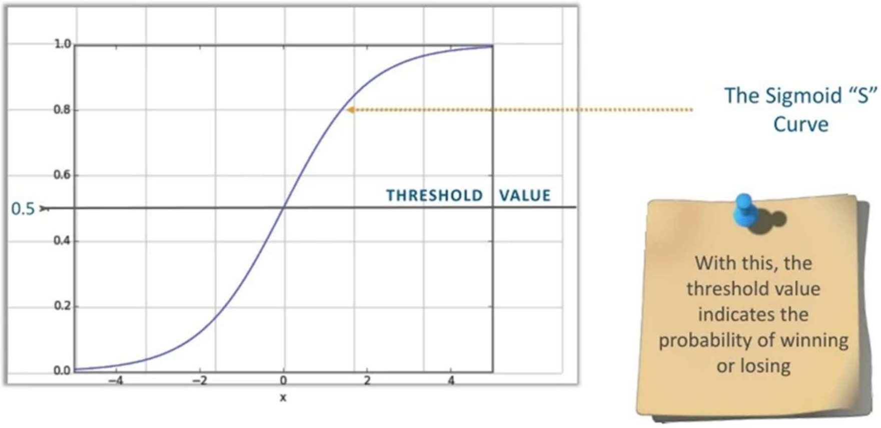


- Predicts the probability of a binary outcome (Yes/No, 0/1)
- Uses the sigmoid function to map inputs to probabilities (0 to 1)
- Ideal for classification tasks

Logistic Regression

Sigmoid Function

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It gives an 'S' shaped curve that can take any real-valued number and map it into a value between 0 and 1. The threshold value defines the probability of either 0 or 1. Such as values above the threshold value tend to 1, and a value below the threshold value tends to 0.



The Sigmoid "S" Curve

With this, the threshold value indicates the probability of winning or losing

In logistic regression, we use a threshold value, usually 0.5, to decide the class label. If the sigmoid output is the same or above the threshold, the input is classified as Class 1. If it is below the threshold, the input is classified as Class 0.

This approach helps to transform continuous input values into meaningful class predictions.

Log-Odds (Logit) Function -

It is the natural logarithm of the odds. Odds are the chances of success divided by the chances of failure and are represented in the form of a ratio. In logistic regression, the log odds of the dependent variable are modeled as a linear combination of the independent variables and the intercept. It is the ratio of the probability of an event occurring to the probability of it not occurring.

Mathematically,

$$\log\text{ (odds) } = \boxed{\log\left(\frac{p}{1-p}\right)}$$

“logit function”

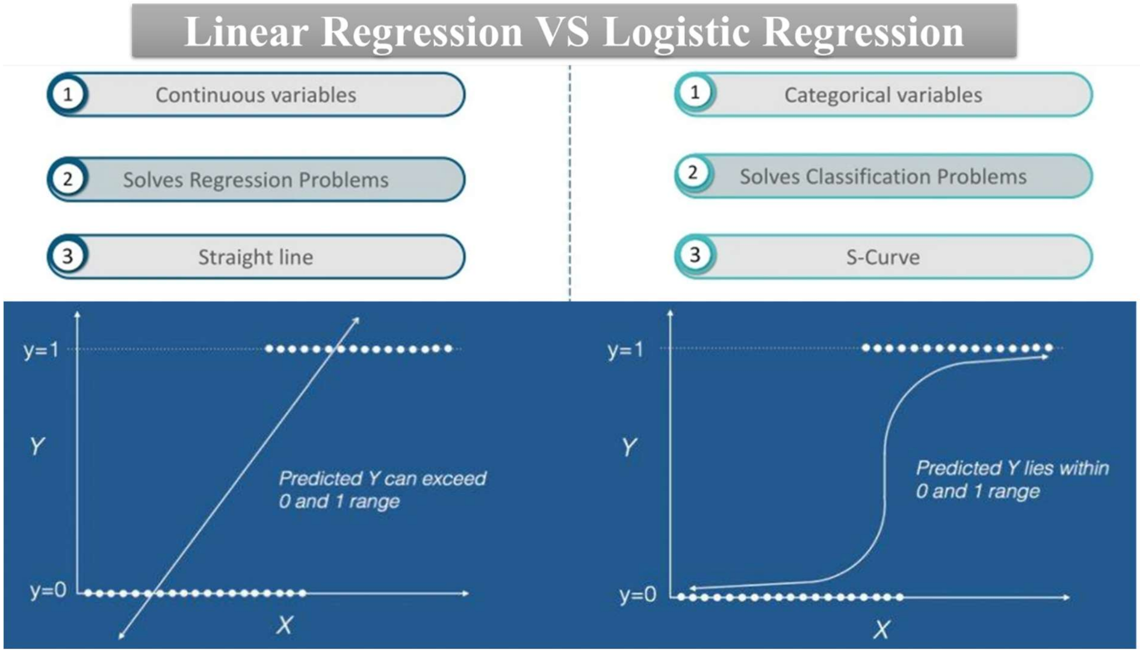
The log of the ratio of the probabilities

→ *The basis of logistic regression*

Linear Regression and Logistics Regression

Logistic regression is similar to the linear regression except how they are used, such as Linear regression is used to solve the regression problem and predict continuous values, whereas Logistic regression is used to solve the Classification problem and usedto predict the discrete values.

Instead of fitting the best fit line, it forms an S-shaped curve that lies between 0 and 1.The S-shaped curve is also known as a logistic function that uses the concept of the threshold. Any value above the threshold will tend to 1, and below the threshold will tend to 0.



Linear Regression VS Logistic Regression

1 Continuous variables

2 Solves Regression Problems

3 Straight line

1 Categorical variables

2 Solves Classification Problems

3 S-Curve

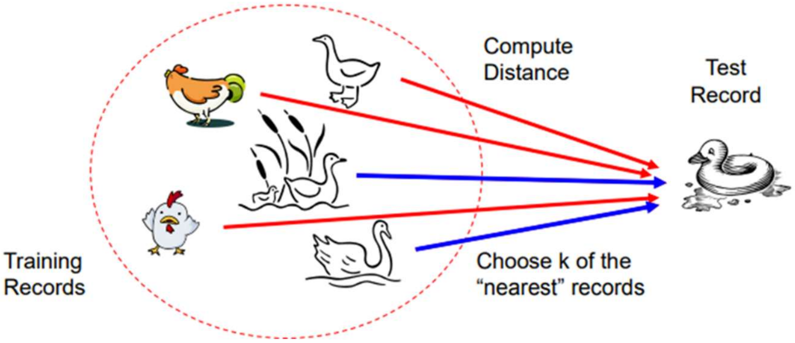
y=1
Y
Predicted Y can exceed 0 and 1 range
y=0
X
y=1
Y
Predicted Y lies within 0 and 1 range
y=0
X

K-Nearest Neighbor (KNN)

Nearest Neighbor Classification

Nearest neighbor classifiers are defined by their characteristic of classifying unlabeled examples by assigning them the class of the most similar labeled examples. Despite the simplicity of this idea, nearest neighbor methods are extremely powerful. They have been used successfully for computer vision applications, recommendation systems, identifying patterns.

Basic idea: If it walks like a duck, quacks like a duck, then it's probably a duck

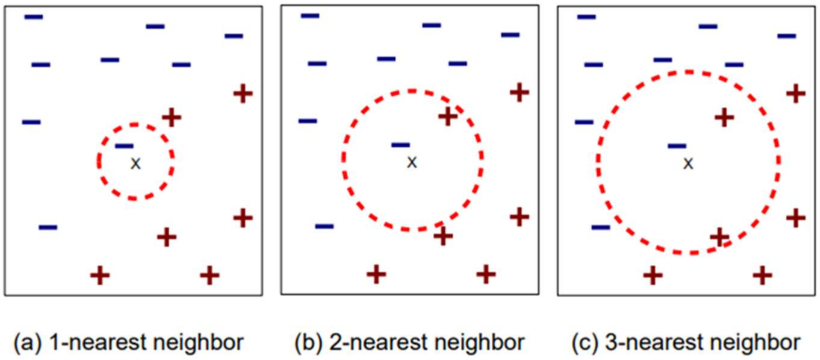


A Nearest Neighbor Classifier requires three things:

- The set of stored records
- Distance Metric to compute distance between records
- The value of k, the number of nearest neighbors to retrieve

To classify an unknown record -

- Compute distance to other training records
- Identify the k nearest neighbors
- Use class labels of nearest neighbors to determine the class label of an unknown record (e.g., by taking a majority vote)



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

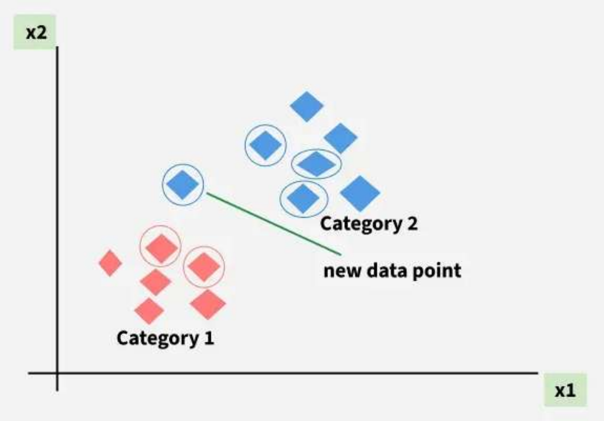
K-nearest neighbors of a record x are data points that have the k smallest distances to x

K Nearest Neighbor (KNN) Algorithm

This algorithm assumes the similarity between the new data and available cases and puts the new case into the category that is most similar to the available categories. KNN is a non-parametric algorithm, which means it does not make any assumption on the underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead, it stores the dataset, and at the time of classification, it acts on the dataset.

The KNN algorithm at the training phase just stores the dataset, and when it gets new data, it classifies that data into a category that is similar to the new data. For each record in the test dataset, KNN identifies k records in the training data that are the "nearest" in similarity, where k is an integer specified in advance. The unlabeled test instance is assigned the class of the majority of the k nearest neighbors.

For example, consider the following table of data points containing two features:



KNN Algorithm working visualization

The new point is classified as Category 2 because most of its closest neighbors are blue squares. KNN assigns the category based on the majority of nearby points. The image shows how KNN predicts the category of a new data point based on its closest neighbours. The red diamonds represent Category 1 and the blue squares represent Category 2. The new data point checks its closest neighbors (circled points). Since the majority of its closest neighbors are blue squares (Category 2) KNN predicts the new data point belongs to Category 2.

KNN works by using proximity and majority voting to make predictions.

'K' in K Nearest Neighbour

In the k-Nearest Neighbours algorithm k is just a number that tells the algorithm how many nearby points or neighbors to look at when it makes a decision.

Example: Imagine you're deciding which fruit it is based on its shape and size. You compare it to fruits you already know.

- If k = 3, the algorithm looks at the 3 closest fruits to the new one.
- If 2 of those 3 fruits are apples and 1 is a banana, the algorithm says the new fruit is an apple because most of its neighbors are apples.

Choosing the appropriate 'k' for the KNN Algorithm -

The value of k in KNN decides how many neighbors the algorithm looks at when making a prediction. Choosing the right k is important for good results. If the data has lots of noise or outliers, using a larger k can make the predictions more stable. But if k is too large, the model may become too simple and miss important patterns and this is called underfitting. So k should be picked carefully based on the data.

Calculating Distance

Deciding how many neighbors to use for kNN determines how well the model will generalize to future data. The algorithm requires a **distance function or a formula** that measures the similarity between two instances. Traditionally, the kNN algorithm uses **Euclidean distance**, which is the distance one would measure if they use a ruler to connect two points.

Euclidean Distance – $D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Minkowsky: $$D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{m} |x_i - y_i|^r\right)^{1/r}$$

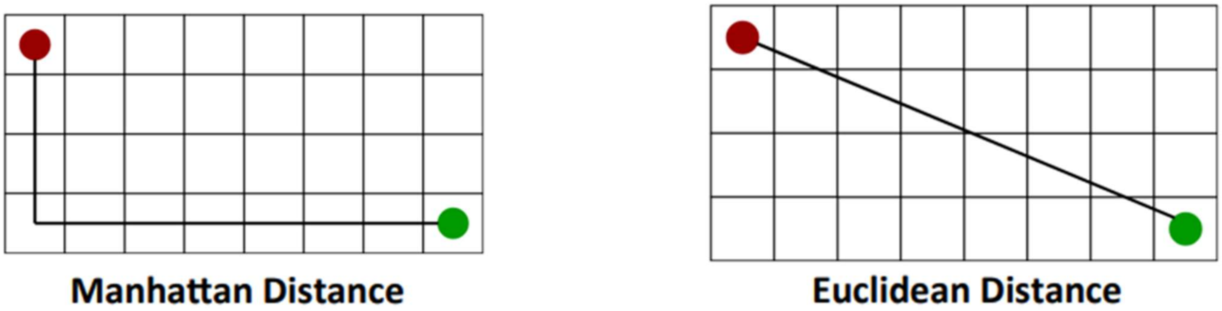
Euclidean: $$D(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

Manhattan / city-block: $$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

Camberra: $$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|}$$

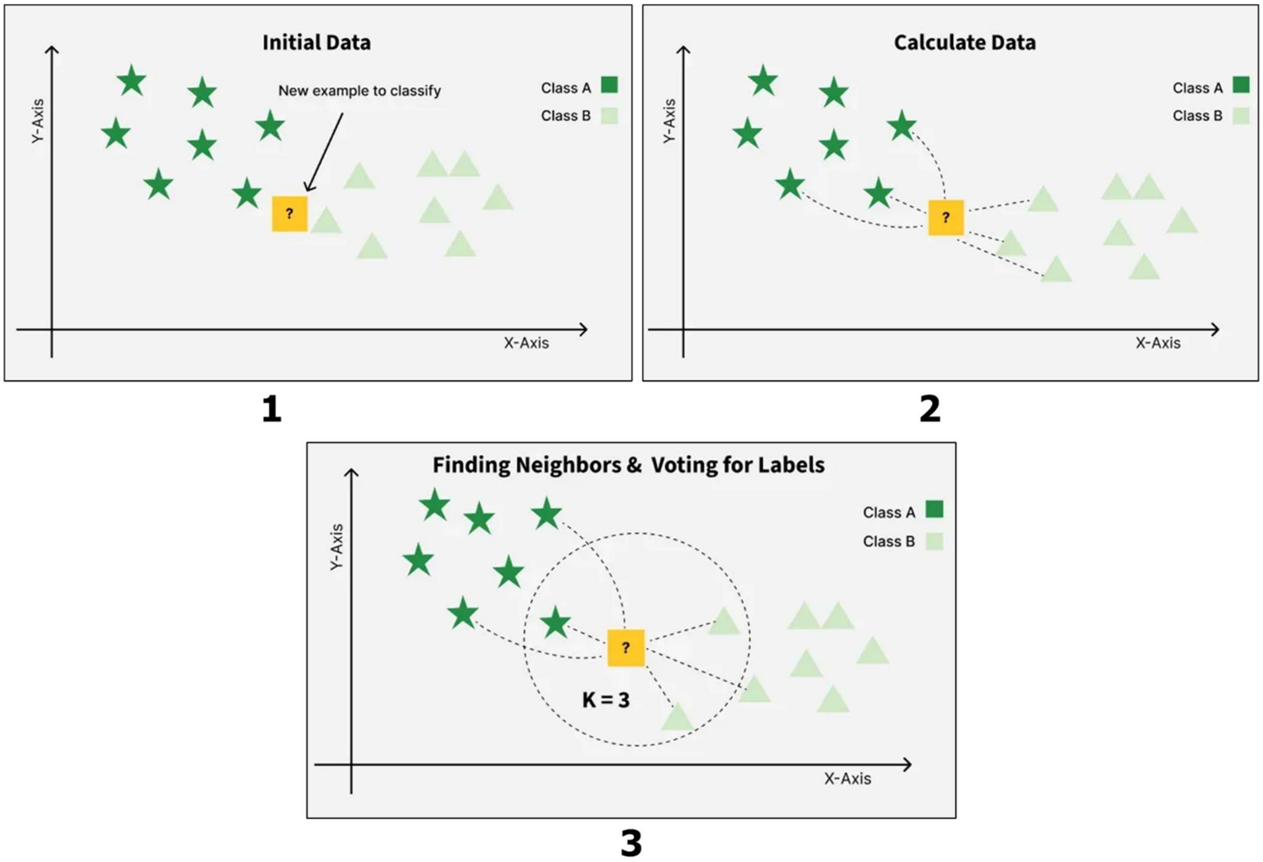
Chebychev: $$D(\boldsymbol{x},\boldsymbol{y}) = \max_{i=1}^{m} |x_i - y_i|$$

Here, ***x*** and ***y*** are the vectors of ***m*** attribute values



Different Equations of selected distance functions

Working of KNN algorithm



Advantages of KNN

- **Simple to use**: Easy to understand and implement.
- **No training step**: No need to train as it just stores the data and uses it during prediction.
- **Few parameters**: Only needs to set the number of neighbors (k) and a distance method.
- **Versatile**: Works for both classification and regression problems.

Disadvantages of KNN

- **Slow with large data**: Needs to compare every point during prediction.
- **Struggles with many features**: Accuracy drops when data has too many features.
- **Can Overfit**: It can overfit especially when the data is high-dimensional or not clean.