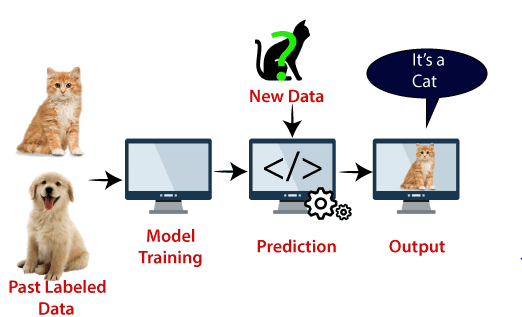
# SUPERVISED MACHINE LEARNING

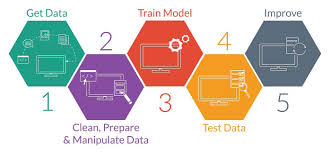
**Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output.**

**The labelled data means some input data is already tagged with the correct output.**

**In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly.**

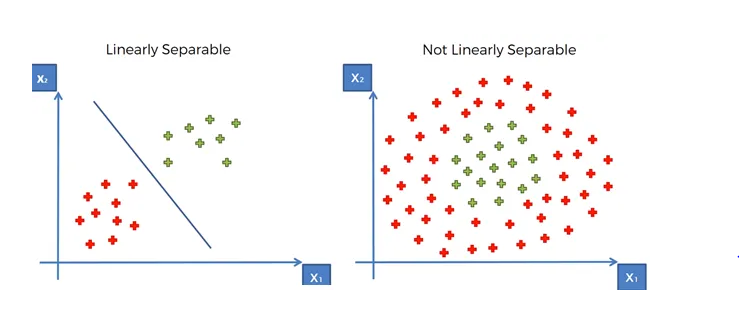
**It applies the same concept as a student learns in the supervision of the teacher.**





# SUPPORT VECTOR MACHINES

# What is SVM?

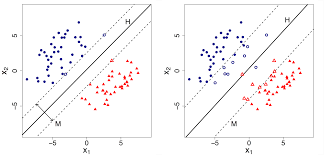


**Support Vector Machines (SVMs)** are a powerful and versatile supervised machine learning algorithm primarily used for classification and regression tasks. They are particularly effective in high-dimensional spaces and are widely used due to their ability to handle complex data relationships.

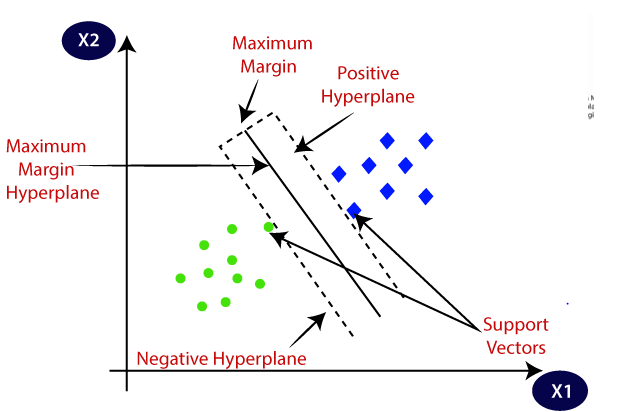
1. **Linear SVM**:
   * It works by identifying a straight hyperplane to separate linearly separable data.
2. **Non-linear SVM**:
   * Uses **kernel functions** to transform non-linear data into a higher-dimensional space where it becomes linearly separable.
   * Common kernel types include:
     + **Linear Kernel**: For linearly separable data.
     + **Polynomial Kernel**: Maps input space into a polynomial feature space.
     + **Radial Basis Function (RBF) Kernel**: Uses Gaussian functions to handle more complex data structures.
     + **Sigmoid Kernel**: Similar to neural networks, used for specific use cases.

**Soft Margin vs. Hard Margin**:

* + **Hard Margin**: Requires perfect separation of classes (used for noise-free data).
  + **Soft Margin**: Allows some misclassification by introducing a penalty term (useful for noisy data).

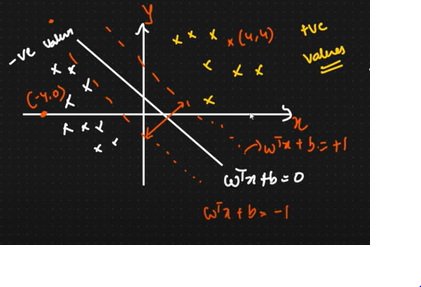


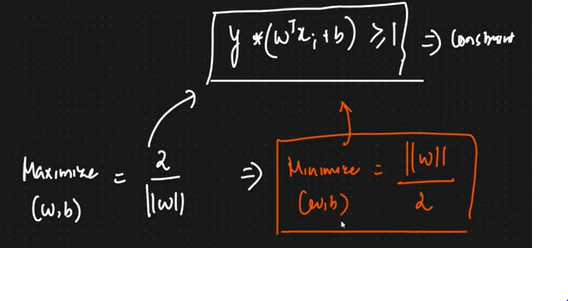
# Working Principle of SVM



The primary objective of the **SVM algorithm** is to identify the **optimal hyperplane** in an N-dimensional space that can effectively separate data points into different classes in the feature space. The algorithm ensures that the margin between the closest points of different classes, known as **support vectors**, is maximized.

The dimension of the hyperplane depends on the number of features. For instance, if there are two input features, the hyperplane is simply a line, and if there are three input features, the hyperplane becomes a 2-D plane. As the number of features increases beyond three, the complexity of visualizing the hyperplane also increases





For soft margin the cost function is



Where e-summation of distance from the misclassified point from the hyperplane

and c-for how many points we can avoid misclassification

# How is SVM Used?

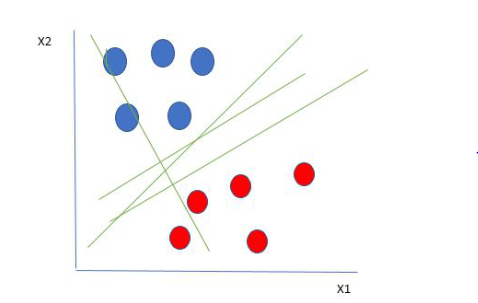
**Algorithm for Implementing SVM**

1. **Import Necessary Libraries**:
   * Import libraries for data handling, model building, and evaluation.
2. **Load and Preprocess the Dataset**:
   * Load the dataset (e.g., from a CSV file or scikit-learn datasets).
   * Split the dataset into features (X) and labels (y).
   * Normalize or standardize the data if required.
3. **Choose and Configure the SVM Model**:
   * Select the kernel type (linear, rbf, poly, or sigmoid).
4. **Train the SVM Model**:
   * Fit the model using the training data.
5. **Make Predictions**:
   * Use the trained model to predict labels for the test data.
6. **Evaluate the Model**:
   * Measure performance using metrics like accuracy, precision.

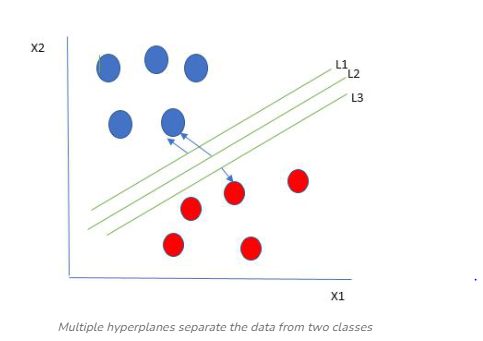
# EXAMPLE

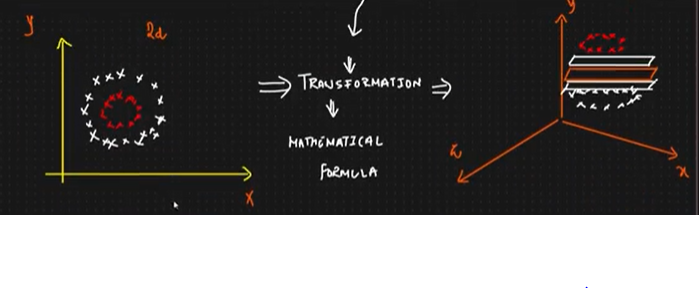
Consider two independent variables, **x1** and **x2**, and one dependent variable represented as either a blue circle or a red circle.

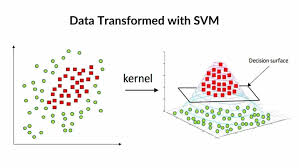
* In this scenario, the hyperplane is a line because we are working with two features (**x1** and **x2**).
* There are multiple lines (or **hyperplanes**) that can separate the data points.
* The challenge is to determine the **best hyperplane** that maximizes the separation margin between the red and blue circles.



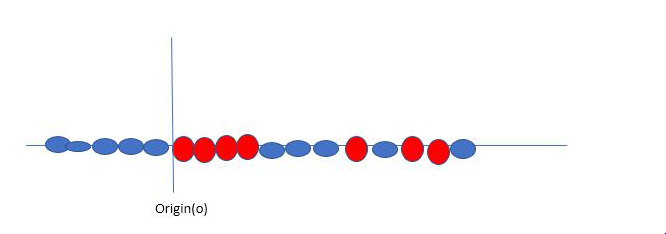
So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the **maximum-margin hyperplane/hard margin**



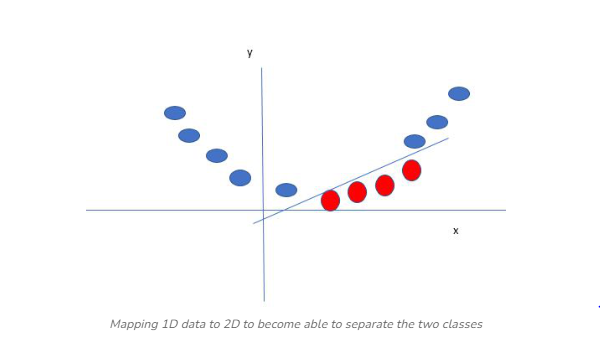




our data is shown in the figure above. SVM solves this by creating a new variable using a **kernel**. We call a point xi on the line and we create a new variable yi as a function of distance from origin o.so if we plot this we get something like as shown below



In this case, the new variable y is created as a function of distance from the origin. A non-linear function that creates a new variable is kernel.

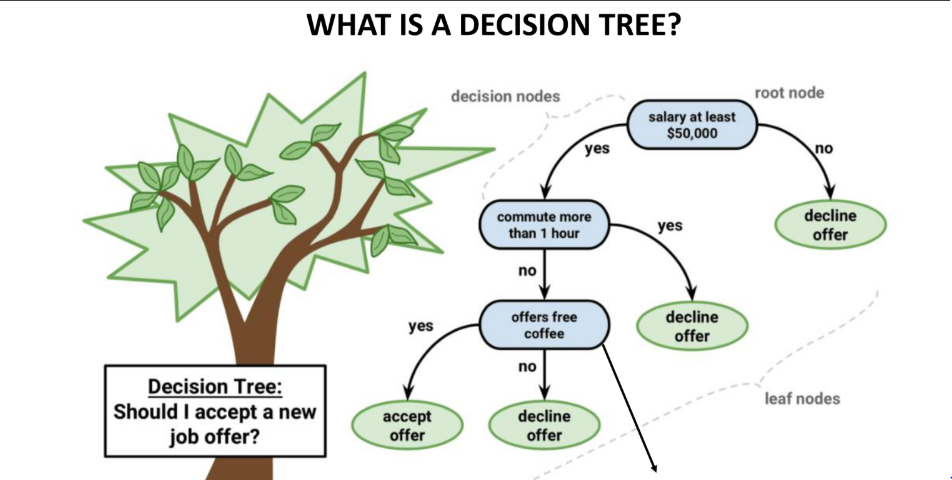


# Applications

1. **Text Classification**:
   * Use SVM to classify emails as spam or not.
2. **Image Classification**:
   * Identify objects in images (e.g., cats vs. dogs).
3. **Fraud Detection**:
   * Detect anomalous transactions.

# 

# DECISION TREE

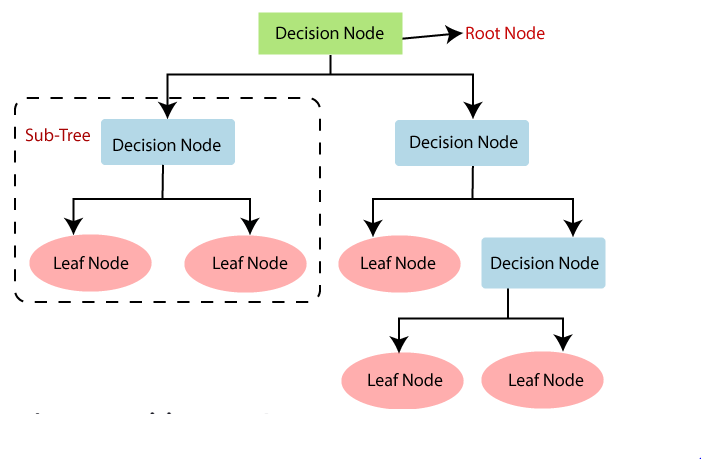


A **Decision Tree** is a supervised machine learning algorithm used for both classification and regression tasks. It is a tree-like structure where decisions are made by splitting data based on conditions. Each internal node represents a test on a feature, each branch represents the outcome of the test, and each leaf node represents a decision or a predicted output.

**Key Components of a Decision Tree**

1. **Root Node**:
   * The topmost node that represents the entire dataset and is split into branches.
2. **Internal Nodes**:
   * Nodes within the tree where data is split based on a feature and a condition (e.g., "Age > 30").
3. **Branches**:
   * Paths from one node to another that represent possible outcomes of a decision rule.
4. **Leaf Nodes**:
   * Terminal nodes of the tree that contain the final output (a class label for classification or a value for regression).
5. **Splitting**:
   * The process of dividing a node into two or more sub-nodes based on specific criteria.

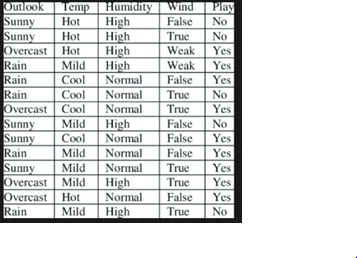
**How Does a Decision Tree Work?**

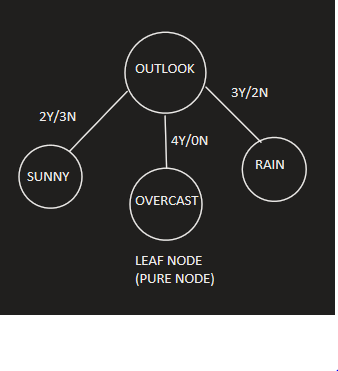


1.**Data Splitting**:

* + The algorithm splits the dataset into subsets based on the most significant feature (one that provides the best separation of classes or reduces variance the most).

1. **Selection Criteria for Splitting**:
   * **Classification Trees**:
     + Use measures like **Gini Impurity** or **Information Gain** (based on Entropy) to determine the best split.
   * **Regression Trees**:
     + Use criteria like **Mean Squared Error (MSE)** or **Mean Absolute Error (MAE)** to determine the best split.
2. **Recursive Process**:
   * The splitting process is repeated recursively on each subset until:
     + A stopping criterion is met (e.g., maximum depth or minimum samples per leaf).
     + Further splitting does not significantly improve the results.
3. **Decision Making**:
   * For **classification**, the majority class in a leaf node becomes the prediction.
   * For **regression**, the mean (or another metric) of the target values in a leaf node is used as the prediction.





## Metrics for Splitting

 Information Gain:

* Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
* It calculates how much information a feature provides us about a class.
* According to the value of information gain, we split the node and build the decision tree.
* A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

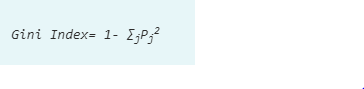
### 

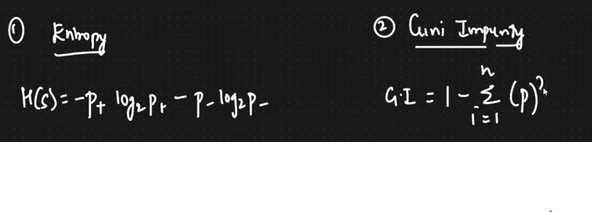
### **Entropy:** Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

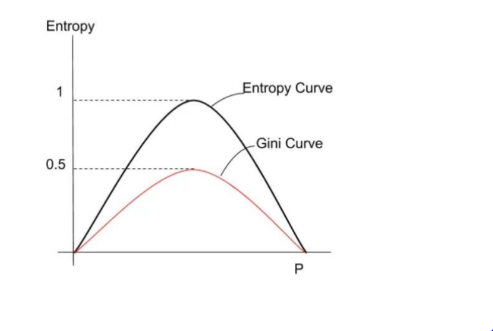
### 

### 2. Gini Index:

* Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
* An attribute with the low Gini index should be preferred as compared to the high Gini index.
* It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
* Gini index can be calculated using the below formula:







ALGORITHM

* **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
* **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
* **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
* **Step-4:** Generate the decision tree node, which contains the best attribute.
* **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

**Example:** Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:

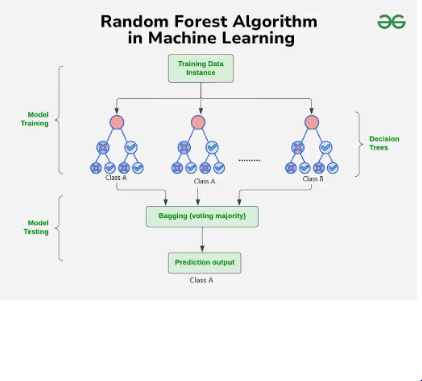


**Advantages of Decision Trees**

1. **Easy to Interpret**:
2. **Handles Categorical and Numerical Data**:
3. **Non-Linear Relationships**:
4. **Feature Selection**:

RANDOM FOREST

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset***.***Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.The greater number of trees in the forest leads to higher accuracy and prevents problem of overfitting.

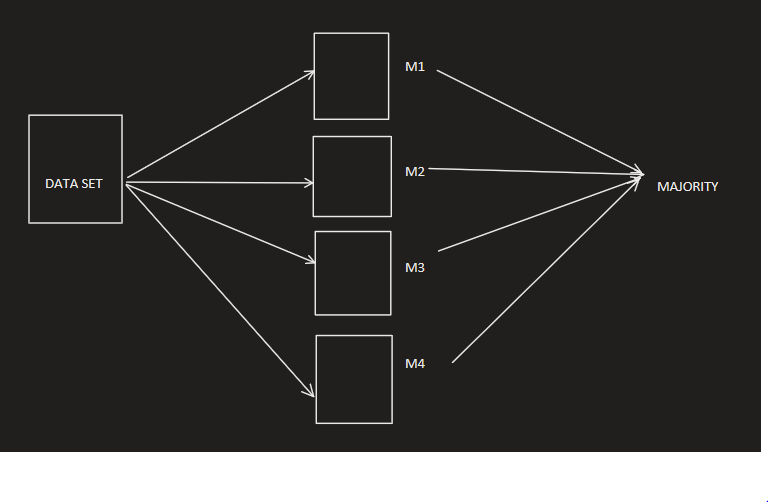


WORKING

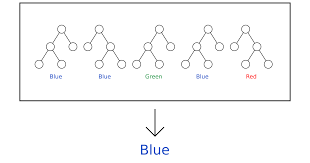
**Working of Random Forest Algorithm**

* **Create Many Decision Trees:** The algorithm makes many [decision trees](https://www.geeksforgeeks.org/machine-learning/decision-tree/) each using a random part of the data. So every tree is a bit different.
* **Pick Random Features:** When building each tree it doesn’t look at all the features (columns) at once. It picks a few at random to decide how to split the data. This helps the trees stay different from each other.
* **Each Tree Makes a Prediction:** Every tree gives its own answer or prediction based on what it learned from its part of the data.
* **Combine the Predictions:**
  + For **classification** we choose a category as the final answer is the one that most trees agree on i.e majority voting.
  + For **regression** we predict a number as the final answer is the average of all the trees predictions.
* **Why It Works Well:** Using random data and features for each tree helps avoid overfitting and makes the overall prediction more accurate and trustworthy.

**How Random Forest Algorithm Works?**

****

* Random Forest builds **multiple decision trees using random samples of the data**. **Each tree is trained on a different subset of the data which makes each tree unique**.

1. When creating each tree the **algorithm randomly selects a subset of features or variables to split the data rather than using all available features at a time. This adds diversity to the trees.**
2. Each decision tree in the forest **makes a prediction based on the data it was trained on. When making final prediction random forest combines the results from all the trees.**
   * For classification tasks the final prediction is decided by a majority vote. This means that the category predicted by most trees is the final prediction.
   1. For regression tasks the final prediction is the average of the predictions from all the trees.
3. The **randomness in data samples and feature selection helps to prevent the model from overfitting making the predictions more accurate and reliable.**
4. 

**Random Forest Algorithm**

***Algorithm Steps***

Implementing the Random Forest algorithm involves several key steps, beginning with data preparation and culminating in model evaluation. The process can be summarized as follows:

**Data Preparation**:

* Clean the dataset by handling missing values and outliers.
* Split the dataset into training and testing subsets.

**Bootstrapping**:

* Generate multiple random samples (subsets) from the training data with replacement.

**Building Decision Trees**:

* For each subset, construct a decision tree. At each node, randomly select a subset of features to determine the best split, enhancing diversity among trees.

**Aggregation**:

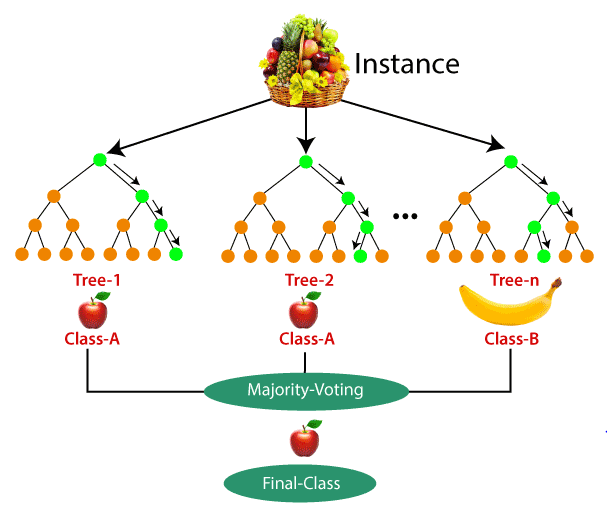
* Combine the predictions from all decision trees to make final predictions:
  + For classification, use majority voting.
  + For regression, compute the average of the predictions.

**Model Evaluation**:

* Assess the model's performance using metrics such as accuracy, precision, recall, and mean squared error on the testing dataset.

Key Challenges

* Time-consuming process: Since random forest algorithms can handle large data sets, they can be provide more accurate predictions, but can be slow to process data as they are computing data for each individual decision tree.
* Requires more resources: Since random forests process larger data sets, they’ll require more resources to store that data.
* More complex: The prediction of a single decision tree is easier to interpret when compared to a forest of them.

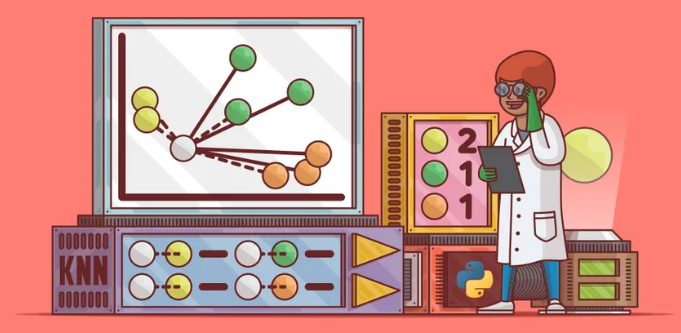


**K NEAREST NEIGHBOUR:**

K-Nearest Neighbour (KNN) is a **supervised machine learning algorithm** used for **classification** and **regression** tasks. It is based on the principle that similar data points exist close to each other in the feature space. KNN is a **non-parametric** and **instance-based** learning method, meaning it does not make assumptions about the data distribution and does not build a model during training.

In KNN, a data point is classified or its value is predicted by considering the **K closest** neighbours in the training dataset, where closeness is determined using a distance metric such as **Euclidean Distance**. The majority class of the neighbours determines the class of the query point (for classification), or the average of their values gives the output (for regression).

KNN is simple, easy to implement, and effective, especially for small to medium datasets. However, it can become computationally expensive for large datasets as it requires calculating distances for every query.



CLASSIFICATION (PREDICTING CLASSES):

1. ***Input***: Training dataset with labelled data points

A black rectangular object with white lines

AI-generated content may be incorrect.

Thus, has fixed number of categories –: 1 or 0

A screen shot of a computer

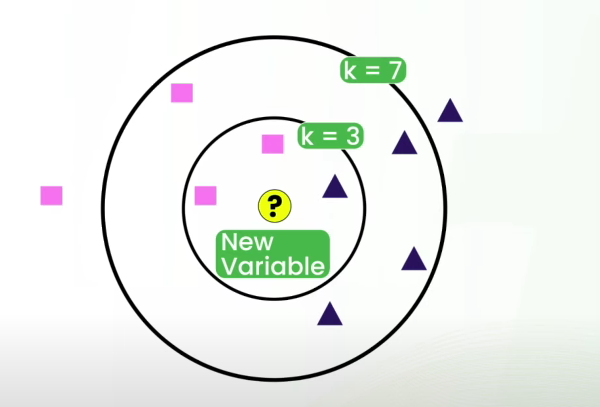
AI-generated content may be incorrect.

The read data is plotted onto a graph as such

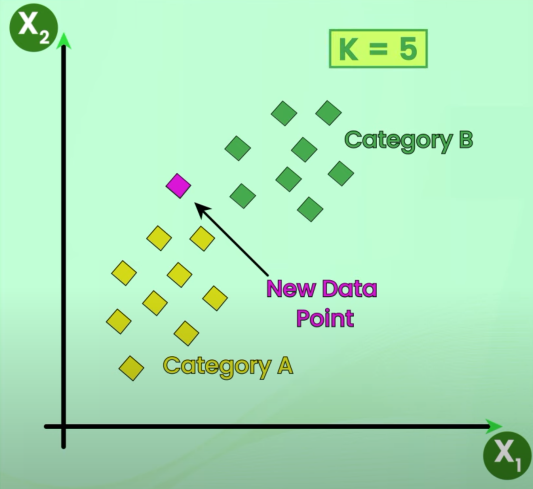
1. ***Choosing the value of k*:** choosing the correct value of k is very important as different values of k may produce different results

As in the following below image (next page) we can see that taking k=3 shows us a majority of squares class but, for the same image if we take k=7 we can observe a majority of triangle class.

The method of selecting the right value of k is called parameter tuning and usually we take the nearest odd value to the square root of ‘n’ as ‘k’, where n is the total number of data points.

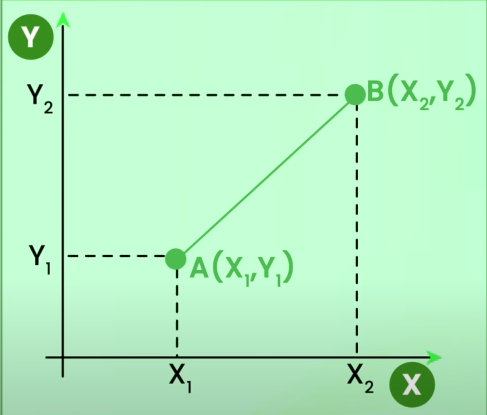


1. ***Calculating the distance***: In order to find the k nearest data points, we need to measure their distances with the new data point. There are many methods to calculate the distance, such as Manhattan or Euclidian distance. Here we will use the Euclidian distance method.

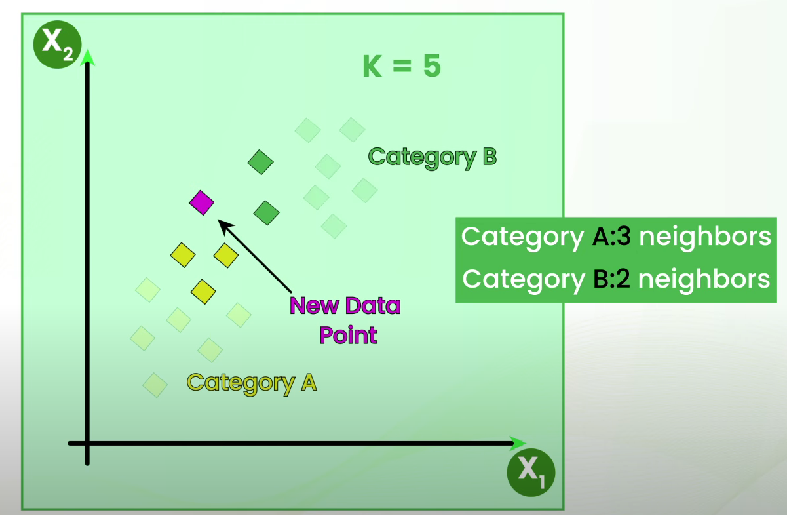


Euclidean Distance: Finding distance between pt 1 (x1, y1) and pt 2 (x2, y2) :-

√ [(x2 – x1)2 + (y2 – y1)2].



1. ***Determining the category:*** we then observe the number of surrounding categories

******

Hence, the category of the new date input is 3.

* REGression

In **K-Nearest Neighbor (KNN) Regression**, the algorithm predicts the output value for a new data point by taking the **average** (or sometimes weighted average) of the values of its **K nearest neighbors**. Unlike KNN classification, where the output is a class label, KNN regression provides a **continuous numeric value** as the prediction.

***Prediction formula :***

A black background with white letters and numbers

AI-generated content may be incorrect.

* Advantages :

1.Extremely Simple and intuitive.

2.Does not require a training phase (quick setup).

3.Works well with small to medium-sized datasets.

* LIMITS :

1. Computationally expensive for large datasets.

2. Sensitive to irrelevant or unscaled features.

3. Performance depends on the choice of **K**.

APPLICATIONS

***Classification:***

1. Image Recognition
2. Text Categorization
3. Medical Diagnosis

***Regression:***

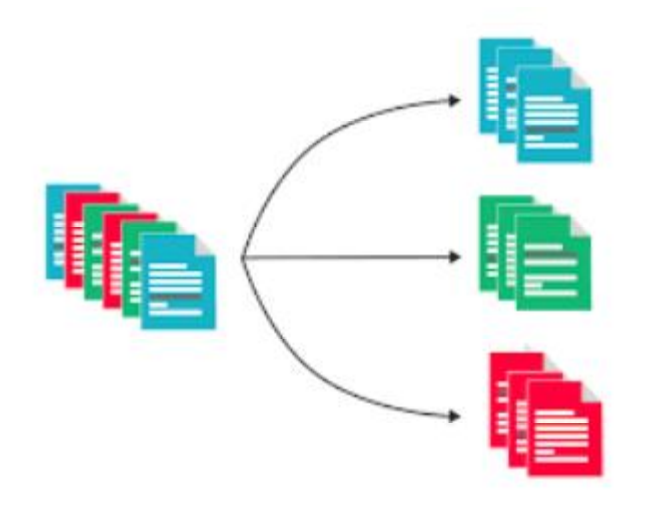
1. Stock Market Analysis
2. Weather Forecasting
3. House Price Prediction

NAIVE BAYES

Naive Bayes classifiers are supervised machine learning algorithms used for classification tasks, based on Bayes Theorem to find probabilities.

INTRODUCTION :

Naïve Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. In other words it is named as “Naive” because it assumes the presence of one feature does not affect other features



BAYES THEOREMFOR NAÏVEBAYES:

Bayes Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes’ theorem is stated mathematically as the following equation:

A black background with white text

AI-generated content may be incorrect.

where X and Y are events and P(X) ≠ 0 where, P(y|X) is Posterior probability:

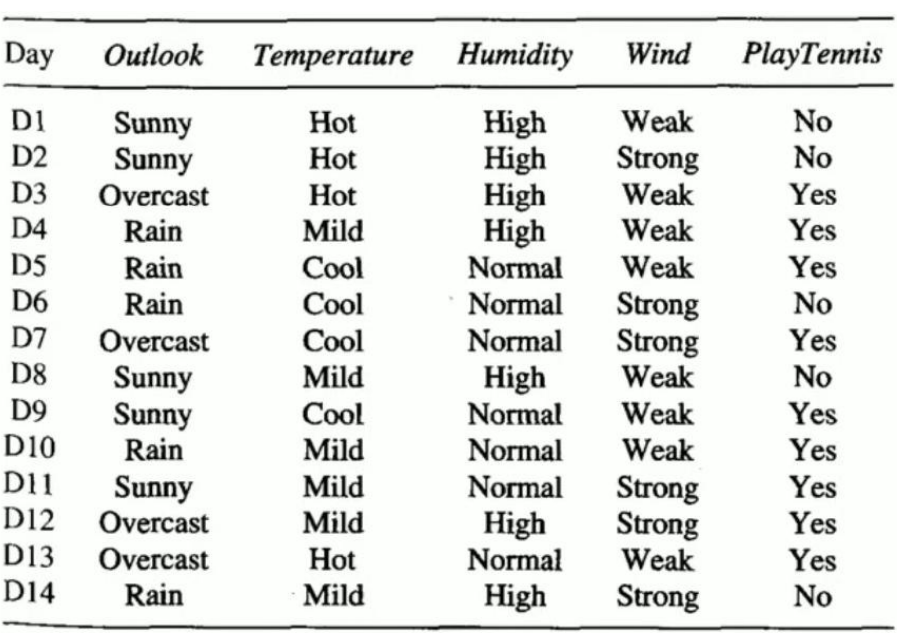
Probability of hypothesis A on the observed event B. P(A|y) is Likelihood probability:

Probability of the evidence given that the probability of a hypothesis is true. X=(x1,x2,x3,…..,xn)

A diagram of a classifier

AI-generated content may be incorrect.

WORKING :



Let us consider the following fictional data sets that describes the weather conditions for playing tennis represented in a tabular format.

We can see

A black text with black letters

AI-generated content may be incorrect.

Now, according to our data sets we can apply bayes theorem in this following way

A chart with different colored squares

AI-generated content may be incorrect.

So, if X = (Rainy, Hot, High, False) y = No So basically, P(y∣X) here means, the probability of “Not playing golf”given that the weather conditions are “Rainy outlook”, “Temperature is hot”, “high humidity” and “no wind”. Now, required instance : Outlook = (sunny ,cool ,high ,strong) Required formula :

A close up of a text

AI-generated content may be incorrect.

So now, we find the value of vnb for yes and no respectively and the one with the greater value is the answer.

A close-up of a text

AI-generated content may be incorrect.

As we can see, the value of Vnb(no) is greater than the valueof yes.Therefore, our final answer is no.

**LINEAR REGRESSION AND LOGISTIC REGRESSION**

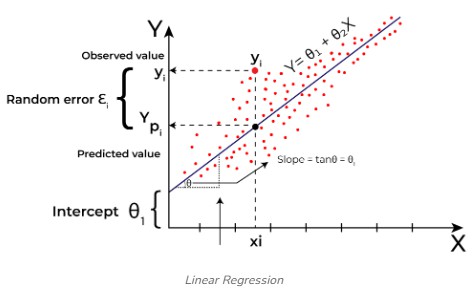
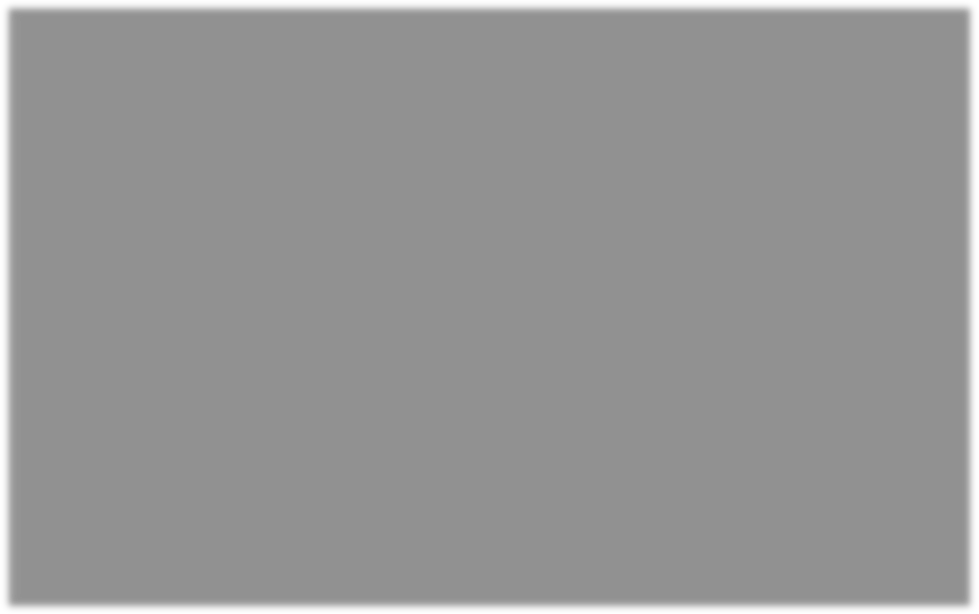
Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. It computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation with observed data.

For example if we want to predict house price we consider various factor such as house age, distance from the main road, location, area and number of room, linear regression uses all these parameter to predict house price as it consider a linear relation between all these features and price of house.

**WORKING PRINCIPLE**

Our primary objective while using linear regression is to locate the best-fit line, which implies that the error between the predicted and actual values should be kept to a minimum. There will be the least error in the best-fit line.

The best Fit Line equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable(s)



Here Y is called a dependent or target variable and X is called an independent variable also known as the predictor of Y. Here, X may be a single feature or multiple features representing the problem.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x)). In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best-fit line for our model.

**Types of Linear Regression**

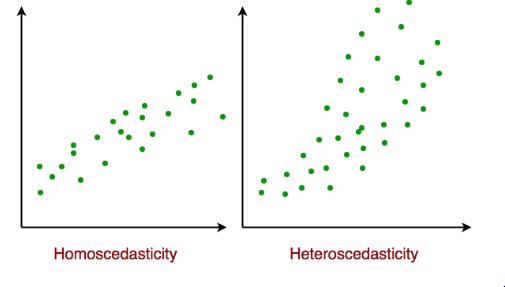
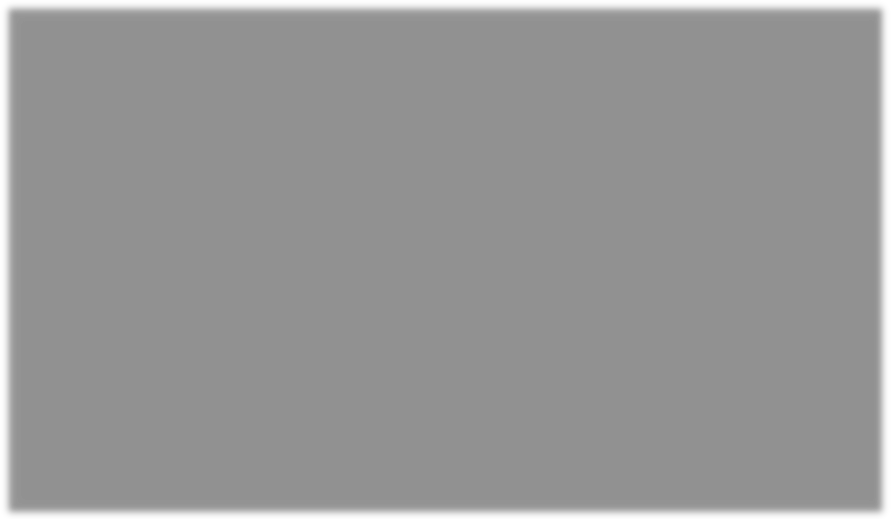
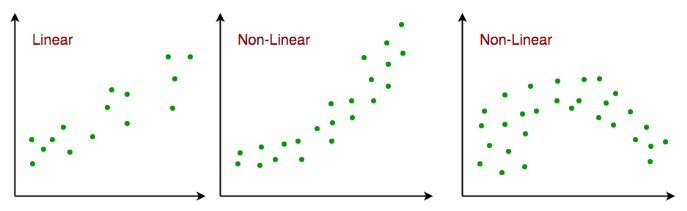
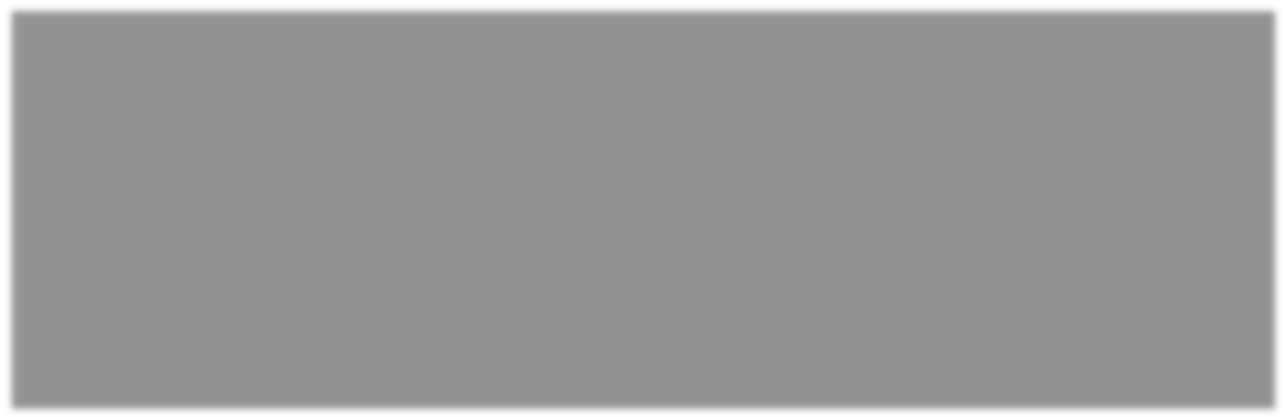
[Simple linear regression](https://www.geeksforgeeks.org/simple-linear-regression-in-python/) is the simplest form of linear regression and it involves only one independent variable and one dependent variable. The equation for simple linear regression is

A white text with black text

AI-generated content may be incorrect.

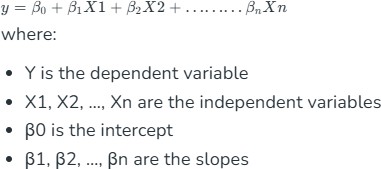
**Assumptions of Simple Linear Regression**

* **Linearity**: The independent and dependent variables have a linear relationship with one another



**2. Multiple Linear Regression**

[Multiple linear regression](https://www.geeksforgeeks.org/ml-multiple-linear-regression-using-python/) involves more than one independent variable and one dependent variable. The equation for multiple linear regression is:



***The*** goal of the algorithm is to find the **best Fit Line** equation that can predict the values based on the independent variables.

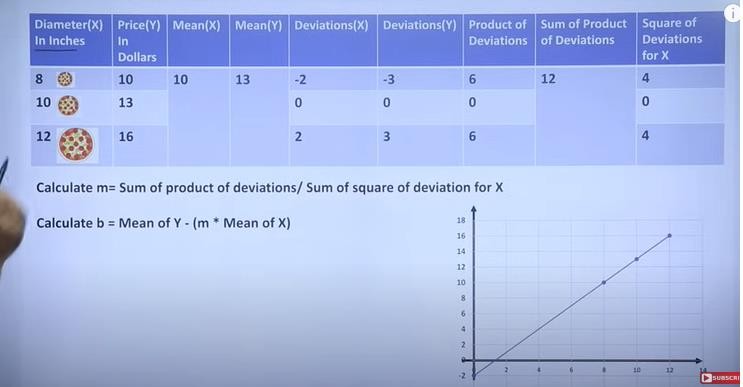
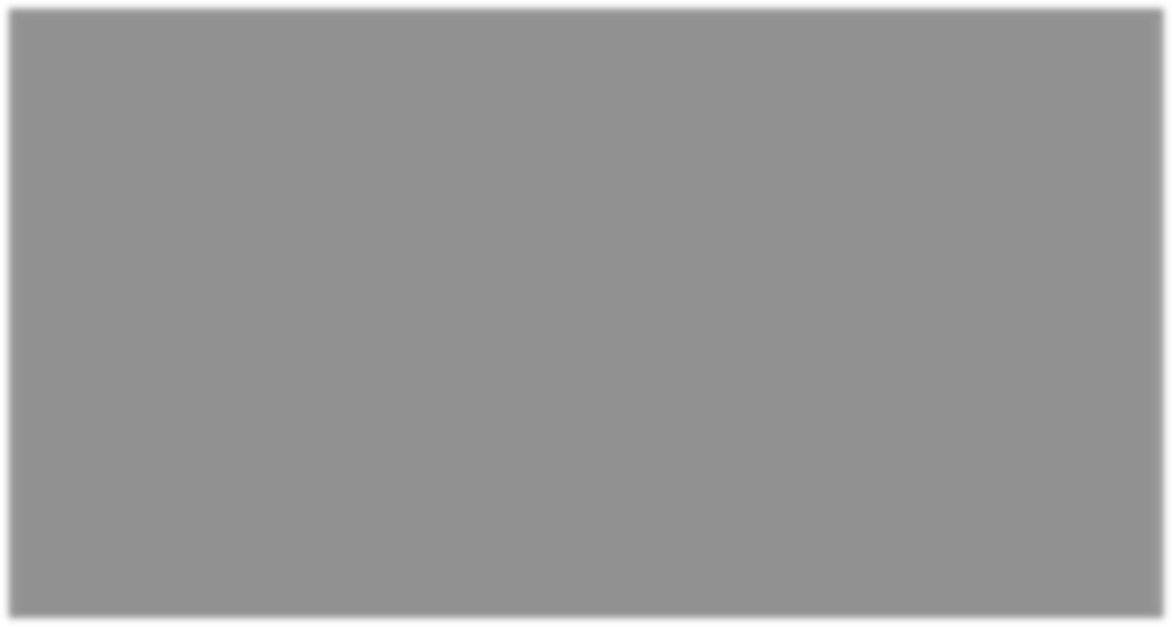
**Assumptions of Multiple Linear Regression**

**No multicollinearity**: There is no high correlation between the independent variables.

**Additivity:** The model assumes that the effect of changes in a predictor variable on the response variable is consistent regardless of the values of the other variables.

**Feature Selection:** In multiple linear regression, it is essential to carefully select the independent variables that will be included in the model.

* **Overfitting:** Overfitting occurs when the model fits the training data too closely, capturing noise or random fluctuations that do not represent the true underlying relationship between variables. **Real Estate Pricing:** In real estate MLR is used to predict property prices based on multiple factors such as location, size, number of bedrooms, etc. This helps buyers and sellers understand market trends and set competitive prices.



**COST FUNCTION**

In linear regression, the cost function used to optimize model parameters is usually the **Mean Squared Error (MSE) or a scaled version of it**. The most common cost function is:

A black background with white numbers and symbols

AI-generated content may be incorrect.

This is just MSE with an extra factor of 1/2 for mathematical convenience when computing gradients during optimization (it simplifies differentiation).

**GRADIENT DESCENT**



A mathematical equation on a black background

AI-generated content may be incorrect.A linear regression model can be trained using the optimization algorithm gradient descent by iteratively modifying the model’s parameters to reduce the mean squared error (MSE) of the model on a training dataset. To update θ1 and θ2 values in order to reduce the Cost function (minimizing RMSE value) and achieve the best-fit line the model uses Gradient Descent. The idea is to start with random θ1 and θ2 values and then iteratively update the values, reaching minimum cost.

α is the learning rate (controls step size)

Now, this value needs to be updated for both θ1 and θ2 until convergence.

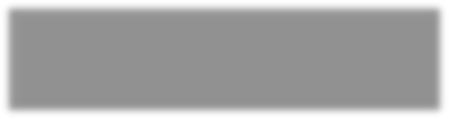
A black background with white text

AI-generated content may be incorrect.

**Evaluation Metrics for Linear Regression** **MSE**

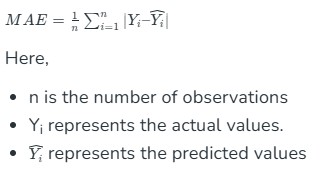
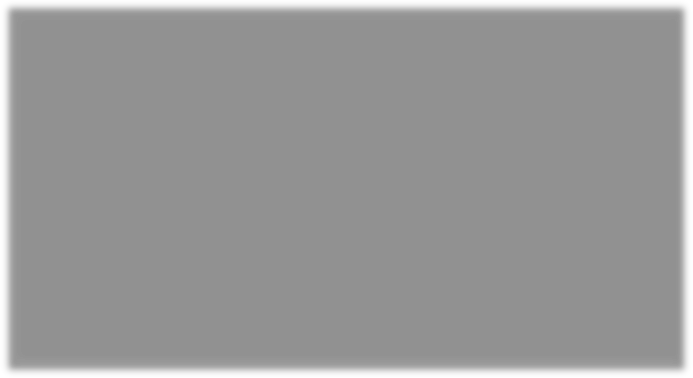
is an evaluation metric that calculates the average of the squared differences

between the actual and predicted values for all the data points. The difference is squared to ensure that negative and positive differences don’t cancel each other out.



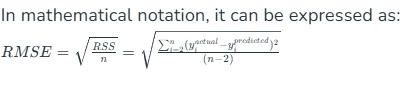
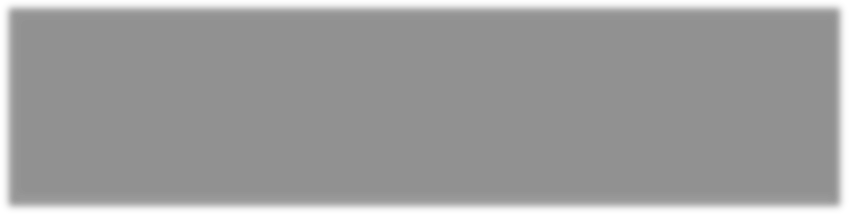
**Mean Absolute Error (MAE)**

[Mean absolute error](https://www.geeksforgeeks.org/how-to-calculate-mean-absolute-error-in-python/) 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.



**Root Mean Squared Error (RMSE)**

The square root of the residuals’ variance is the [Root Mean Squared Error](https://www.geeksforgeeks.org/root-mean-square-error-in-r-programming/). It describes how well the observed data points match the expected values, or the model’s absolute fit to the data.



**What is Logistic Regression?**

Logistic regression is used for binary [classification](https://www.geeksforgeeks.org/getting-started-with-classification/) where we use [sigmoid function](https://www.geeksforgeeks.org/derivative-of-the-sigmoid-function/), that takes input as independent variables and produces a probability value between 0 and 1.

For example, we have two classes Class 0 and Class 1 if the value of the logistic function for an input is greater than 0.5 (threshold value) then it belongs to Class 1 otherwise it belongs to Class 0. It’s referred to as regression because it is the extension of [linear regression](https://www.geeksforgeeks.org/ml-linear-regression/) but is mainly used for classification problems.

**Key Points:**

* Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.
* It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
* In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).

**Logistic Function – Sigmoid Function**

* It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this

limit, so it forms a curve like the “S” form.

* The S-form curve is called the Sigmoid function or the logistic function.

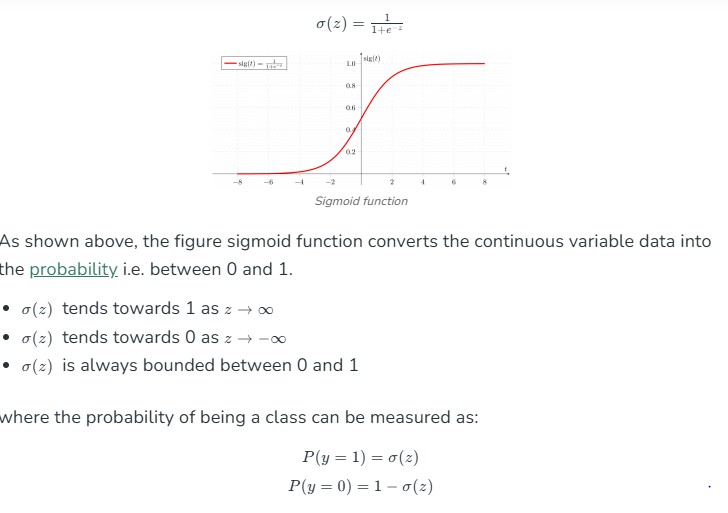
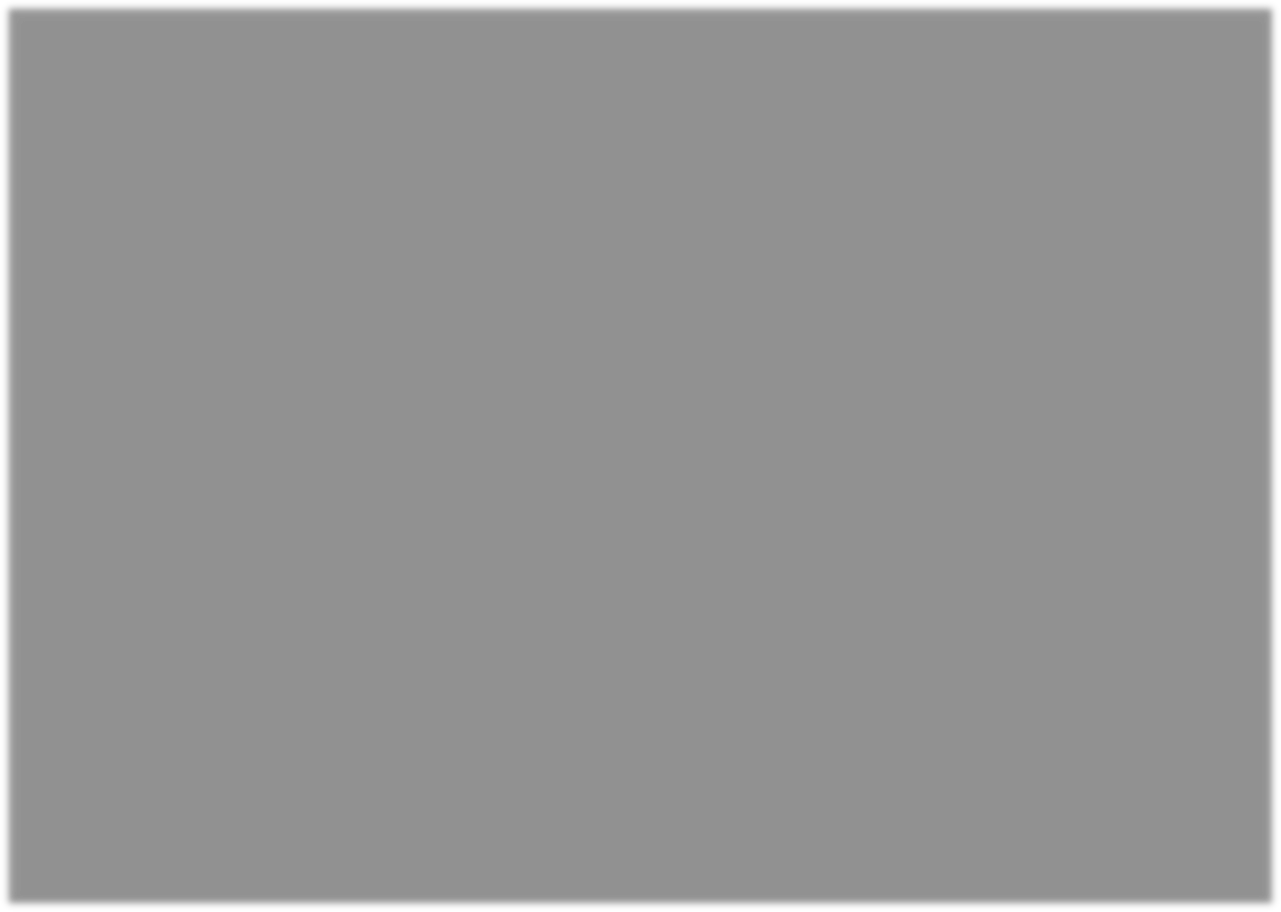
**Types of Logistic Regression**

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

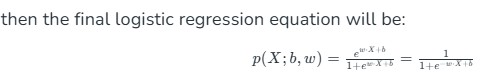
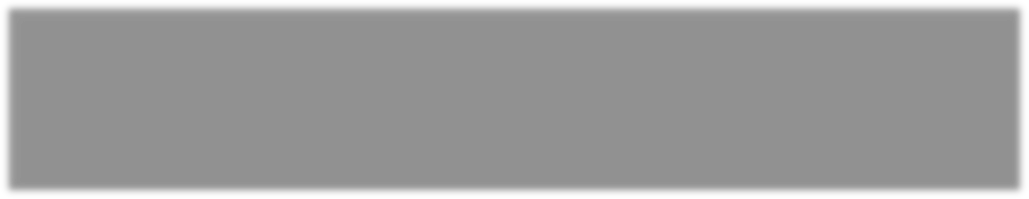
1. **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
2. **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as “cat”, “dogs”, 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”.

**Sigmoid Function**

Now we use the [sigmoid function](https://www.geeksforgeeks.org/derivative-of-the-sigmoid-function/) where the input will be z and we find the probability between 0 and 1. i.e. predicted y.



**Logistic Regression Equation**



**Differences Between Linear and Logistic Regression**

|  |  |
| --- | --- |
| **Linear Regression** | **Logistic Regression** |
| Linear regression is used to predict the continuous dependent variable using a given set of independent variables. | Logistic regression is used to predict the categorical dependent variable using a given set of independent variables. |

|  |  |
| --- | --- |
| **Linear Regression** | **Logistic Regression** |
| Linear regression is used for solving regression problem. | It is used for solving classification problems. |
| In this we predict the value of continuous variables | In this we predict values of categorical variables |
| In this we find best fit line. | In this we find S-Curve. |
| Least square estimation method is used for estimation of accuracy. | Maximum likelihood estimation method is used for Estimation of accuracy. |
| The output must be continuous value, such as price, age, etc. | Output must be categorical value such as 0 or 1, Yes or no, etc. |
| It required linear relationship between dependent and independent variables. | It not required linear relationship. |
| There may be collinearity between the independent variables. | There should be little to no collinearity between independent variables. |