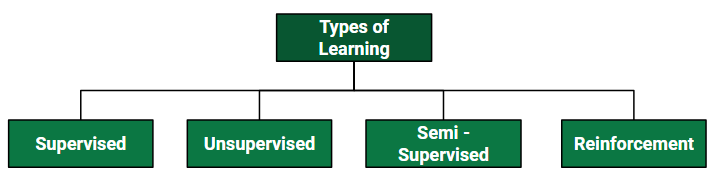
**Supervised Machine Learning**

**Supervised Machine Learning**

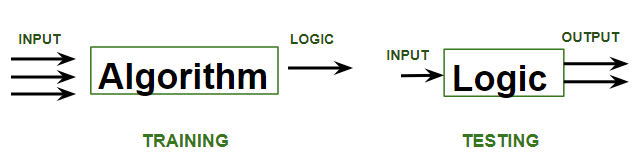
A machine is said to be learning from **past Experiences**(data feed-in) with respect to some class of**tasks** if its **Performance** in a given Task improves with the Experience. For example, assume that a machine has to predict whether a customer will buy a specific product let’s say “Antivirus” this year or not. The machine will do it by looking at the **previous knowledge/past experiences**i.e. the data of products that the customer had bought every year and if he buys an Antivirus every year, then there is a high probability that the customer is going to buy an antivirus this year as well. This is how machine learning works at the basic conceptual level.



## Supervised Machine Learning

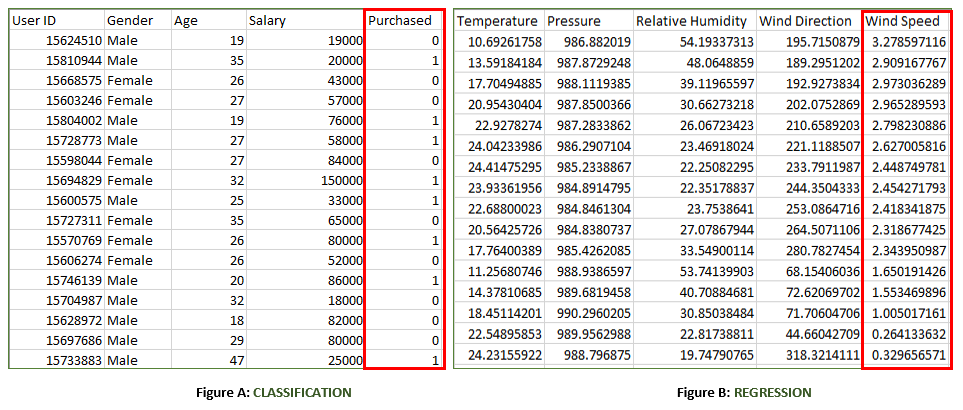
Supervised learning is a machine learning technique that is widely used in various fields such as finance, healthcare, marketing, and more. It is a form of machine learning in which the algorithm is trained on labeled data to make predictions or decisions based on the data inputs.In supervised learning, the algorithm learns a mapping between the input and output data. This mapping is learned from a labeled dataset, which consists of pairs of input and output data. The algorithm tries to learn the relationship between the input and output data so that it can make accurate predictions on new, unseen data.

Let us discuss what learning for a machine is as shown below media as follows:



**Supervised learning**is where the model is trained on a labelled dataset. A **labelled** dataset is one that has both input and output parameters. In this type of learning both training and validation, datasets are labelled as shown in the figures below.

The labeled dataset used in supervised learning consists of input features and corresponding output labels. The input features are the attributes or characteristics of the data that are used to make predictions, while the output labels are the desired outcomes or targets that the algorithm tries to predict.



Both the above figures have labelled data set as follows:

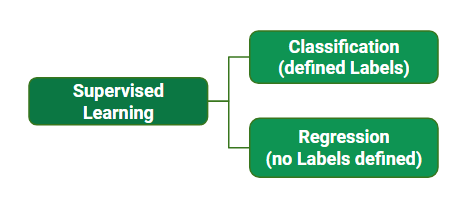
* **Figure A:**It is a dataset of a shopping store that is useful in predicting whether a customer will purchase a particular product under consideration or not based on his/ her gender, age, and salary.   
  **Input:** Gender, Age, Salary   
  **Output:** Purchased i.e. 0 or 1; 1 means yes the customer will purchase and 0 means that the customer won’t purchase it.
* **Figure B:**It is a Meteorological dataset that serves the purpose of predicting wind speed based on different parameters.   
  **Input:** Dew Point, Temperature, Pressure, Relative Humidity, Wind Direction   
  **Output:** Wind Speed

**Training the system:** While training the model, data is usually split in the ratio of 80:20 i.e. 80% as training data and the rest as testing data. In training data, we feed input as well as output for 80% of data. The model learns from training data only. We use different machine learning algorithms(which we will discuss in detail in the next articles) to build our model. Learning means that the model will build some logic of its own.   
Once the model is ready then it is good to be tested. At the time of testing, the input is fed from the remaining 20% of data that the model has never seen before, the model will predict some value and we will compare it with the actual output and calculate the accuracy.

## **Types of Supervised Learning Algorithm**

Supervised learning is typically divided into two main categories: regression and classification. In regression, the algorithm learns to predict a continuous output value, such as the price of a house or the temperature of a city. In classification, the algorithm learns to predict a categorical output variable or class label, such as whether a customer is likely to purchase a product or not.

One of the primary advantages of supervised learning is that it allows for the creation of complex models that can make accurate predictions on new data. However, supervised learning requires large amounts of labeled training data to be effective. Additionally, the quality and representativeness of the training data can have a significant impact on the accuracy of the model.  
Supervised learning can be further classified into two categories:



### **Regression**

Regression is a supervised learning technique used to predict continuous numerical values based on input features. It aims to establish a functional relationship between independent variables and a dependent variable, such as predicting house prices based on features like size, bedrooms, and location.  
The goal is to minimize the difference between predicted and actual values using algorithms like Linear Regression, Decision Trees, or Neural Networks, ensuring the model captures underlying patterns in the data.

### **Classification**

Classification is a type of supervised learning that categorizes input data into predefined labels. It involves training a model on labeled examples to learn patterns between input features and output classes. In classification, the target variable is a categorical value. For example, classifying emails as spam or not.  
The model’s goal is to generalize this learning to make accurate predictions on new, unseen data. Algorithms like Decision Trees, Support Vector Machines, and Neural Networks are commonly used for classification tasks.

**NOTE:** There are common Supervised Machine Learning Algorithm that can be used for both regression and classification task.

## **Supervised Machine Learning Algorithm**

Supervised learning can be further divided into several different types, each with its own unique characteristics and applications. Here are some of the most common types of supervised learning algorithms:

* [**Linear Regression**](https://www.geeksforgeeks.org/ml-linear-regression/): Linear regression is a type of regression algorithm that is used to predict a continuous output value. It is one of the simplest and most widely used algorithms in supervised learning. In linear regression, the algorithm tries to find a linear relationship between the input features and the output value. The output value is predicted based on the weighted sum of the input features.
* [**Logistic Regression**](https://www.geeksforgeeks.org/understanding-logistic-regression/): Logistic regression is a type of classification algorithm that is used to predict a binary output variable. It is commonly used in machine learning applications where the output variable is either true or false, such as in fraud detection or spam filtering. In logistic regression, the algorithm tries to find a linear relationship between the input features and the output variable. The output variable is then transformed using a logistic function to produce a probability value between 0 and 1.
* [**Decision Trees**](https://www.geeksforgeeks.org/decision-tree/): Decision tree is a tree-like structure that is used to model decisions and their possible consequences. Each internal node in the tree represents a decision, while each leaf node represents a possible outcome. Decision trees can be used to model complex relationships between input features and output variables.  
  A decision tree is a type of algorithm that is used for both classification and regression tasks.
  + **Decision Trees Regression:**Decision Trees can be utilized for regression tasks by predicting the value linked with a leaf node.
  + **Decision Trees Classification:**Random Forest is a machine learning algorithm that uses multiple decision trees to improve classification and prevent overfitting.
* [**Random Forests**](https://www.geeksforgeeks.org/random-forest-regression-in-python/): Random forests are made up of multiple decision trees that work together to make predictions. Each tree in the forest is trained on a different subset of the input features and data. The final prediction is made by aggregating the predictions of all the trees in the forest.  
  Random forests are an ensemble learning technique that is used for both classification and regression tasks.
  + **Random Forest Regression**: It combines multiple decision trees to reduce overfitting and improve prediction accuracy.
  + **Random Forest Classifier:** Combines several decision trees to improve the accuracy of classification while minimizing overfitting.
* [**Support Vector Machine(SVM)**](https://www.geeksforgeeks.org/support-vector-machine-algorithm/): The SVM algorithm creates a hyperplane to segregate n-dimensional space into classes and identify the correct category of new data points. The extreme cases that help create the hyperplane are called support vectors, hence the name Support Vector Machine.  
  A Support Vector Machine is a type of algorithm that is used for both classification and regression tasks
  + **Support Vector Regression:**It is a extension of Support Vector Machines (SVM) used for predicting continuous values.
  + **Support Vector Classifier:**It aims to find the best hyperplane that maximizes the margin between data points of different classes.
* [**K-Nearest Neighbors**](https://www.geeksforgeeks.org/k-nearest-neighbours/)**(KNN)**: KNN works by finding k training examples closest to a given input and then predicts the class or value based on the majority class or average value of these neighbors. The performance of KNN can be influenced by the choice of k and the distance metric used to measure proximity. However, it is intuitive but can be sensitive to noisy data and requires careful selection of k for optimal results.  
  A K-Nearest Neighbors (KNN) is a type of algorithm that is used for both classification and regression tasks.
  + **K-Nearest Neighbors Regressio**n: It predicts continuous values by averaging the outputs of the k closest neighbors.
  + **K-Nearest Neighbors Classification:**Data points are classified based on the majority class of their k closest neighbors.
* [**Gradient Boosting**](https://www.geeksforgeeks.org/ml-gradient-boosting/): Gradient Boosting combines weak learners, like decision trees, to create a strong model. It iteratively builds new models that correct errors made by previous ones. Each new model is trained to minimize residual errors, resulting in a powerful predictor capable of handling complex data relationships.  
  A Gradient Boosting is a type of algorithm that is used for both classification and regression tasks.
  + **Gradient Boosting Regression:** It builds an ensemble of weak learners to improve prediction accuracy through iterative training.
  + **Gradient Boosting Classification:**Creates a group of classifiers to continually enhance the accuracy of predictions through iterations

## A**dvantages of Supervised Learning**

The power of supervised learning lies in its ability to accurately predict patterns and make data-driven decisions across a variety of applications. Here are some advantages listed below:

1. Labeled training data benefits supervised learning by enabling models to accurately learn patterns and relationships between inputs and outputs.
2. Supervised learning models can accurately predict and classify new data.
3. Supervised learning has a wide range of applications, including classification, regression, and even more complex problems like image recognition and natural language processing.
4. Well-established evaluation metrics, including accuracy, precision, recall, and F1-score, facilitate the assessment of supervised learning model performance.

## **Disadvantages of Supervised Learning**

Although supervised learning methods have benefits, their limitations require careful consideration during problem formulation, data collection, model selection, and evaluation. Here are some disadvantages listed below:

1. [**Overfitting**](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/)**:**Models can overfit training data, which leads to poor performance on new, unseen data due to the capture of noise.
2. [**Feature Engineering**](https://www.geeksforgeeks.org/what-is-feature-engineering/)**:**Extracting relevant features from raw data is crucial for model performance, but this process can be time-consuming and may require domain expertise.
3. **Bias in Models:** Training data biases can lead to unfair predictions.
4. Supervised learning heavily depends on labeled training data, which can be costly, time-consuming, and may require domain expertise.

**Regression in Machine Learning:**

# Linear Regression in Machine learning

## **What is Linear Regression?**

Linear regression is a type of [supervised machine learning](https://www.geeksforgeeks.org/supervised-machine-learning/) algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data.

When there is only one independent feature, it is known as [Simple Linear Regression](https://www.geeksforgeeks.org/simple-linear-regression-using-r/), and when there are more than one feature, it is known as [Multiple Linear Regression](https://www.geeksforgeeks.org/ml-multiple-linear-regression-using-python/).

Similarly, when there is only one dependent variable, it is considered [Univariate Linear Regression](https://www.geeksforgeeks.org/univariate-linear-regression-in-python/), while when there are more than one dependent variables, it is known as [Multivariate Regression](https://www.geeksforgeeks.org/multivariate-regression/).

## Types of Linear Regression

There are two main types of linear regression:

### **Simple Linear Regression**

This 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:  
𝑦=𝛽0+𝛽1𝑋*y*=*β*0​+*β*1​*X*  
where:

* Y is the dependent variable
* X is the independent variable
* β0 is the intercept
* β1 is the slope

### **Multiple Linear Regression**

This involves more than one independent variable and one dependent variable. The equation for multiple linear regression is:  
𝑦=𝛽0+𝛽1𝑋+𝛽2𝑋+………𝛽𝑛𝑋*y*=*β*0​+*β*1​*X*+*β*2​*X*+………*βn*​*X*  
where:

* Y is the dependent variable
* X1, X2, …, Xp are the independent variables
* β0 is the intercept
* β1, β2, …, βn are the slopes

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

In regression set of records are present with X and Y values and these values are used to learn a function so if you want to predict Y from an unknown X this learned function can be used. In regression we have to find the value of Y, So, a function is required that predicts continuous Y in the case of regression given X as independent features.

## Advantages & Disadvantages of Linear Regression

### Advantages of Linear Regression

* Linear regression is a relatively simple algorithm, making it easy to understand and implement. The coefficients of the linear regression model can be interpreted as the change in the dependent variable for a one-unit change in the independent variable, providing insights into the relationships between variables.
* Linear regression is computationally efficient and can handle large datasets effectively. It can be trained quickly on large datasets, making it suitable for real-time applications.
* Linear regression is relatively robust to outliers compared to other machine learning algorithms. Outliers may have a smaller impact on the overall model performance.
* Linear regression often serves as a good baseline model for comparison with more complex machine learning algorithms.
* Linear regression is a well-established algorithm with a rich history and is widely available in various machine learning libraries and software packages.

### Disadvantages of Linear Regression

* Linear regression assumes a linear relationship between the dependent and independent variables. If the relationship is not linear, the model may not perform well.
* Linear regression is sensitive to multicollinearity, which occurs when there is a high correlation between independent variables. Multicollinearity can inflate the variance of the coefficients and lead to unstable model predictions.
* Linear regression assumes that the features are already in a suitable form for the model. Feature engineering may be required to transform features into a format that can be effectively used by the model.
* Linear regression is susceptible to both overfitting and underfitting. Overfitting occurs when the model learns the training data too well and fails to generalize to unseen data. Underfitting occurs when the model is too simple to capture the underlying relationships in the data.
* Linear regression provides limited explanatory power for complex relationships between variables. More advanced machine learning techniques may be necessary for deeper insights.

# Implementation of Polynomial Regression

**Polynomial Regression**is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modelled as an nth-degree polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y | x). In this article, we’ll go in-depth about polynomial regression.

## **What is a Polynomial Regression?**

* There are some relationships that a researcher will hypothesize is curvilinear. Clearly, such types of cases will include a polynomial term.
* Inspection of residuals. If we try to fit a linear model to curved data, a scatter plot of residuals (Y-axis) on the predictor (X-axis) will have patches of many positive residuals in the middle. Hence in such a situation, it is not appropriate.
* An assumption in the usual multiple linear regression analysis is that all the independent variables are independent. In the [polynomial regression](https://www.geeksforgeeks.org/polynomial-regression-from-scratch-using-python/) model, this assumption is not satisfied

## Why Polynomial Regression?

Polynomial regression is a type of regression analysis used in statistics and machine learning when the relationship between the independent variable (input) and the dependent variable (output) is not linear. While simple linear regression models the relationship as a straight line, polynomial regression allows for more flexibility by fitting a polynomial equation to the data.

When the relationship between the variables is better represented by a curve rather than a straight line, polynomial regression can capture the non-linear patterns in the data

## **Application of Polynomial Regression**

The reason behind the vast use cases of the polynomial regression is that approximately all of the real-world data is non-linear in nature and hence when we fit a non-linear model on the data or a curvilinear regression line then the results that we obtain are far better than what we can achieve with the standard linear regression. Some of the use cases of the Polynomial regression are as stated below:

* The growth rate of tissues.
* Progression of disease epidemics
* Distribution of carbon isotopes in lake sediments

## **Advantages** & **Disadvantages of using Polynomial Regression**

### **Advantages of using Polynomial Regression**

* A broad range of functions can be fit under it.
* Polynomial basically fits a wide range of curvatures.
* Polynomial provides the best approximation of the relationship between dependent and independent variables.

### **Disadvantages of using Polynomial Regression**

* These are too sensitive to outliers.
* The presence of one or two [outliers](https://www.geeksforgeeks.org/machine-learning-outlier/) in the data can seriously affect the results of nonlinear analysis.
* In addition, there are unfortunately fewer model validation tools for the detection of outliers in nonlinear regression than there are for linear regression.

# Stepwise Regression

Stepwise regression is a method of fitting a regression model by iteratively adding or removing variables. It is used to build a model that is accurate and parsimonious, meaning that it has the smallest number of variables that can explain the data.

**There are two main types of**[**stepwise regression**](https://www.geeksforgeeks.org/types-of-regression-techniques/)**:**

* Forward Selection –  In forward selection, the algorithm starts with an empty model and iteratively adds variables to the model until no further improvement is made.
* Backward Elimination – In backward elimination, the algorithm starts with a model that includes all variables and iteratively removes variables until no further improvement is made.

## Use of Stepwise Regression?

The primary use of stepwise regression is to build a regression model that is accurate and parsimonious. In other words, it is used to find the smallest number of variables that can explain the data.

Stepwise regression is a popular method for model selection because it can automatically select the most important variables for the model and build a parsimonious model. This can save time and effort for the data scientist or analyst, who does not have to manually select the variables for the model.

Stepwise regression can also improve the model’s performance by reducing the number of variables and eliminating any unnecessary or irrelevant variables. This can help to prevent overfitting, which can occur when the model is too complex and does not generalize well to new data.

Overall, the use of stepwise regression is to build accurate and parsimonious regression models that can handle complex, non-linear relationships in the data. It is a popular and effective method for model selection in many different domains.

## **Ridge Regression**

Multicollinearity occurs when two or more predictor variables in a regression model are highly correlated, leading to unreliable and unstable estimates of regression coefficients. ***Ridge regression is a procedure for eliminating the bias of coefficients and reducing the mean square error by shrinking the coefficients of a model towards zero in order to solve problems of overfitting or multicollinearity that are normally associated with ordinary least squares regression.***

It adds an additional term to the [OLS](https://www.geeksforgeeks.org/ordinary-least-squares-ols-using-statsmodels/)loss function that pulls the estimating coefficients toward zero.**This is done by adding a penalty term to the log likelihood, where this penalty term is governed by a parameter denoted as lambda (λ), thus lowering the variance of the model and increasing its stability as well as the robustness of the prediction made by the model.**

## Advantages and Disadvantages of Ridge Regression

### Advantages:

* **Stability**: Ridge regression provides more stable estimates in the presence of multicollinearity.
* **Bias-Variance Tradeoff**: By introducing bias, ridge regression reduces the variance of the estimates, leading to lower MSE.
* **Interpretability**: Unlike principal component regression, ridge regression retains the original predictors, making the results easier to interpret.

### Disadvantages:

* **Bias Introduction**: The introduction of bias can lead to underestimation of the true effects of the predictors.
* **Parameter Selection**: Choosing the optimal ridge parameter 𝑘k can be challenging and computationally intensive.
* **Not Suitable for Variable Selection**: Ridge regression does not perform variable selection, meaning all predictors remain in the model, even those with negligible effects.

# What is lasso regression?

## Understanding Lasso Regression

Lasso (Least Absolute Shrinkage and Selection Operator) regression typically belongs to regularization techniques category, which is usually applied to avoid [overfitting](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/). Lasso Regression enhance the [linear regression](https://www.geeksforgeeks.org/ml-linear-regression/) concept by making use of a [regularization](https://www.geeksforgeeks.org/regularization-in-machine-learning/) process in the standard regression equation. Linear Regression operates by minimizing the sum of squared discrepancies between the observed and predicted values by fitting a line (or, in higher dimensions, a plane or hyperplane) to the data points.  
  
However, [multicollinearity](https://www.geeksforgeeks.org/multicollinearity-in-data/) a condition in which features have a strong correlation with one another occurs in real-words datasets. This is when the regularization approach of Lasso Regression comes in handy. Regularization, in simple term add penalty term to model, preventing it from overfitting.

For example, if you’re attempting with a model to forecast house prices based on features such as location, square footage, and the number of bedrooms. Lasso Regression will let us determine which feature is more important or whereas location and square footage are major determinants of price. By zeroing out the coefficient for the bedroom feature, Lasso simplifies the model in order to increase the model accuracy.

## When to use Lasso Regression

When working with high-dimensional datasets that contain a large number of features some of which may be redundant or irrelevant, lasso regression is very helpful. Moreover, we can use lasso regression in following situations:

* **Feature Selection**: By reducing the coefficients of less significant features to zero, Lasso regression automatically chooses a selection of features. When you have a lot of features and want to find the ones that are most significant, this is helpful.
* **Collinearity:** By reducing the coefficients of correlated variables and choosing one, lasso regression might be useful when there is multicollinearity—that is, when the predictor variables have a high degree of correlation with one another.
* **Regularization**: By penalizing big coefficients, Lasso regression can aid in preventing overfitting. When the number of predictors approaches or surpasses the number of observations, this becomes particularly significant.
* **Interpretability**: Compared to conventional linear regression models that incorporate all features, lasso regression often yields sparse models with fewer non-zero coefficients. This could make the final model simpler to understand.

## Advantages of Lasso Regression

* **Feature Selection:** Lasso regression eliminates the need to manually select the most relevant features, hence, the developed regression model becomes simpler and more explainable.
* **Regularization:** Lasso constrains large coefficients, so a less biased model is generated, which is robust and general in its predictions.
* **Interpretability:**With lasso, models are often sparsity induced, therefore, they are easier to interpret and explain, which is essential in fields like health care and finance.
* **Handles Large Feature Spaces:**Lasso lends itself to dealing with high-dimensional data like we have in genomic as well as imaging studies.

## Disadvantages of Lasso Regression

* **Selection Bias:**Lasso, might arbitrarily choose one variable in a group of highly correlated variables rather than the other, thereby yielding a biased model in the end.
* **Sensitive to Scale:** Lasso is demanding in the respect that features of different orders have a tendency to affect the regularization line and the model’s precision.
* **Impact of Outliers:** Lasso can be easily affected by the outliers in the given data, resulting into the overfitting of the coefficients.
* **Model Instability:**In the environment of multiple correlated variables the lasso’s selection of variable may be unstable, which results in different variable subsets each time in tiny data change.
* **Tuning Parameter Selection:** Analyzing different λ (alpha) values may be problematic and maybe solved by cross-validation.

# Bayesian Regression

In Bayesian linear regression, the mean of one parameter is characterized by a weighted sum of other variables. This type of conditional modeling aims to determine the prior distribution of the regressors as well as other variables describing the allocation of the regressand) and eventually permits the out-of-sample forecasting of the regressand conditional on observations of the regression coefficients.

The normal linear equation, where the distribution of display style YY given by display style XX is Gaussian, is the most basic and popular variant of this model. The future can be determined analytically for this model, and a specific set of prior probabilities for the parameters is known as conjugate priors. The posteriors usually have more randomly selected priors.

When the dataset has too few or poorly dispersed data, Bayesian Regression might be quite helpful. In contrast to conventional regression techniques, where the output is only derived from a single number of each attribute, a Bayesian Regression model's output is derived from a probability distribution.

The result, "y," is produced by a normal distribution (where the variance and mean are normalized). The goal of the Bayesian Regression Model is to identify the 'posterior' distribution again for model parameters rather than the model parameters themselves. The model parameters will be expected to follow a distribution in addition to the output y.

The posterior expression is given below:

Posterior = (Likelihood \* Prior)/Normalization

The expression parameters are explained below:

* Posterior: It is the likelihood that an event, such as H, will take place given the occurrence of another event, such as E, i.e., P(H | E).
* Likelihood: It is a likelihood function in which a marginalization parameter variable is used.
* Priority: This refers to the likelihood that event H happened before event A, i.e., P(H) (H)

This is the same as Bayes' Theorem, which states the following -

P(A|B) = (P(B|A) P(A))/P(B)

P(A) is the likelihood that event A will occur, while P(A|B) is the likelihood that event A will occur, provided that event B has already occurred. Here, A and B seem to be events. P(B), the likelihood of event B happening cannot be zero because it already has.

According to the aforementioned formula, we get a prior probability for the model parameters that is proportional to the probability of the data divided by the posterior distribution of the parameters, unlike Ordinary Least Square (OLS), which is what we observed in the case of the OLS.

The value of probability will rise as more data points are collected and eventually surpass the previous value. The parameter values converge to values obtained by OLS in the case of an unlimited number of data points. Consequently, we start our regression method with an estimate (the prior value).

As we begin to include additional data points, the accuracy of our model improves. Therefore, to make a Bayesian Ridge Regression model accurate, a considerable amount of train data is required.

Let's quickly review the mathematical side of the situation now. If 'y' is the expected value in a linear model, then

y(w,x) = w0+w1x1+...+wpxp

where, The vector "w" is made up of the elements w0, w1,... The weight value is expressed as 'x'.

w=(w1…wp)

As a result, the output "y" is now considered to be the Gaussian distribution around Xw for Bayesian Regression to produce a completely probabilistic model, as demonstrated below:

p(y|X, w. 𝛼) = N(y|Xw, 𝛼)

where the Gamma distribution prior hyper-parameter alpha is present. It is handled as a probability calculated from the data. The Bayesian Ridge Regression implementation is provided below.

The Bayesian Ridge Regression formula on which it is based is as follows:

p(y|λ)=N(w|0, λ^-1Ip)

where alpha is the Gamma distribution's shape parameter before the alpha parameter and lambda is the distribution's shape parameter before the lambda parameter.

We have discussed Bayesian Linear Regression so, let us now discuss some of its real-life applications.

## **Advantages Of Bayesian Regression**

Some of the main advantages of Bayesian Regression are defined below:

* Extremely efficient when the dataset is tiny.
* Particularly well-suited for online learning as opposed to batch learning, when we know the complete dataset before we begin training the model. This is so that Bayesian Regression can be used without having to save data.
* The Bayesian technique has been successfully applied and is quite strong mathematically. Therefore, using this requires no additional prior knowledge of the dataset.

Let us now look at some disadvantages of Bayesian Regression.

## **Disadvantages Of Bayesian Regression**

Some common disadvantages of using Bayesian Regression:

* The model's inference process can take some time.
* The Bayesian strategy is not worthwhile if there is a lot of data accessible for our dataset, and the regular probability approach does the task more effectively.

After going through the definitions, applications, and advantages and disadvantages of Bayesian Linear Regression, it is time for us to explore how to implement Bayesian Regression using Python.

# What is Mean Absolute Error (MAE)?

# Mean Absolute Error (MAE) is a simple yet powerful metric used to evaluate the accuracy of regression models. It measures the average absolute difference between the predicted values and the actual target values. Unlike other metrics, MAE doesn’t square the errors, which means it gives equal weight to all errors, regardless of their direction. This property makes MAE particularly useful when you want to understand the magnitude of errors without considering whether they are overestimations or underestimations

# Why MAE Matters?

MAE offers several advantages that make it a valuable tool in assessing model performance:

1. **Robustness to Outliers**: Unlike some other metrics, MAE is less sensitive to extreme values (outliers) in the data. This makes it a suitable choice when your dataset contains outliers that might skew other metrics like Mean Squared Error (MSE).
2. **Interpretability**: MAE is in the same unit as the original target variable, making it easy to interpret. For example, if your model predicts house prices in dollars, the MAE will also be in dollars, providing a tangible understanding of the error magnitude.
3. **Simple and Intuitive**: MAE is straightforward to calculate and understand. Each absolute difference contributes equally to the final score, making it easy to grasp the overall performance of the model.

## What is R-Squared

**R-squared** is a statistical measure that represents the goodness of fit of a regression model. The value of R-square lies between 0 to 1. Where we get R-square equals 1 when the model perfectly fits the data and there is no difference between the predicted value and actual value.  However, we get R-square equals 0 when the model does not predict any variability in the model and it does not learn any relationship between the dependent and independent variables.

# Logistic Regression in Machine Learning

## 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.

## 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”.

## Differences Between Linear and Logistic Regression

The difference between linear regression and logistic regression is that linear regression output is the continuous value that can be anything while logistic regression predicts the probability that an instance belongs to a given class or not.

| **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 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. |

# Decision Tree

## What is a Decision Tree?

A **decision tree**is a flowchart-like structure used to make decisions or predictions. It consists of nodes representing decisions or tests on attributes, branches representing the outcome of these decisions, and leaf nodes representing final outcomes or predictions. Each internal node corresponds to a test on an attribute, each branch corresponds to the result of the test, and each leaf node corresponds to a class label or a continuous value.

## Structure of a Decision Tree

1. **Root Node**: Represents the entire dataset and the initial decision to be made.
2. **Internal Nodes**: Represent decisions or tests on attributes. Each internal node has one or more branches.
3. **Branches**: Represent the outcome of a decision or test, leading to another node.
4. **Leaf Nodes**: Represent the final decision or prediction. No further splits occur at these nodes.

## How Decision Trees Work?

The process of creating a decision tree involves:

1. **Selecting the Best Attribute**: Using a metric like Gini impurity, entropy, or information gain, the best attribute to split the data is selected.
2. **Splitting the Dataset**: The dataset is split into subsets based on the selected attribute.
3. **Repeating the Process**: The process is repeated recursively for each subset, creating a new internal node or leaf node until a stopping criterion is met (e.g., all instances in a node belong to the same class or a predefined depth is reached).

## Metrics for Splitting

* **Gini Impurity**: Measures the likelihood of an incorrect classification of a new instance if it was randomly classified according to the distribution of classes in the dataset.
  + Gini=1–∑𝑖=1𝑛(𝑝𝑖)2Gini=1–∑*i*=1*n*​(*pi*​)2, where pi​ is the probability of an instance being classified into a particular class.
* **Entropy**: Measures the amount of uncertainty or impurity in the dataset.
  + Entropy=−∑𝑖=1𝑛𝑝𝑖log⁡2(𝑝𝑖)Entropy=−∑*i*=1*n*​*pi*​log2​(*pi*​), where pi​ is the probability of an instance being classified into a particular class.
* **Information Gain**: Measures the reduction in entropy or Gini impurity after a dataset is split on an attribute.
  + InformationGain=Entropyparent–∑𝑖=1𝑛(∣𝐷𝑖∣∣𝐷∣∗Entropy(𝐷𝑖))InformationGain=Entropyparent​–∑*i*=1*n*​(∣*D*∣∣*Di*​∣​∗Entropy(*Di*​)), where Di​ is the subset of D after splitting by an attribute.

## Advantages of Decision Trees

* **Simplicity and Interpretability**: Decision trees are easy to understand and interpret. The visual representation closely mirrors human decision-making processes.
* **Versatility**: Can be used for both classification and regression tasks.
* **No Need for Feature Scaling**: Decision trees do not require normalization or scaling of the data.
* **Handles Non-linear Relationships**: Capable of capturing non-linear relationships between features and target variables.

## Disadvantages of Decision Trees

* **Overfitting**: Decision trees can easily overfit the training data, especially if they are deep with many nodes.
* **Instability**: Small variations in the data can result in a completely different tree being generated.
* **Bias towards Features with More Levels**: Features with more levels can dominate the tree structure.

# K-Nearest Neighbor(KNN) Algorithm

## What is the K-Nearest Neighbors Algorithm?

KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the [supervised learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning) domain and finds intense application in pattern recognition, [data mining](https://www.geeksforgeeks.org/data-mining), and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a [Gaussian distribution](https://www.geeksforgeeks.org/mathematics-probability-distributions-set-3-normal-distribution) of the given data). We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute

## Advantages of the KNN Algorithm

* **Easy to implement** as the complexity of the algorithm is not that high.
* **Adapts Easily** – As per the working of the KNN algorithm it stores all the data in memory storage and hence whenever a new example or data point is added then the algorithm adjusts itself as per that new example and has its contribution to the future predictions as well.
* **Few Hyperparameters** – The only parameters which are required in the training of a KNN algorithm are the value of k and the choice of the distance metric which we would like to choose from our evaluation metric.

## Disadvantages of the KNN Algorithm

* **Does not scale** – As we have heard about this that the KNN algorithm is also considered a Lazy Algorithm. The main significance of this term is that this takes lots of computing power as well as data storage. This makes this algorithm both time-consuming and resource exhausting.
* **Curse of Dimensionality** – There is a term known as the peaking phenomenon according to this the KNN algorithm is affected by the [curse of dimensionality](https://www.geeksforgeeks.org/videos/curse-of-dimensionality-in-machine-learning) which implies the algorithm faces a hard time classifying the data points properly when the dimensionality is too high.
* **Prone to Overfitting** – As the algorithm is affected due to the curse of dimensionality it is prone to the problem of overfitting as well. Hence generally [feature selection](https://www.geeksforgeeks.org/feature-selection-techniques-in-machine-learning) as well as [dimensionality reduction](https://www.geeksforgeeks.org/dimensionality-reduction) techniques are applied to deal with this problem

# Naive Bayes Classifiers

## What is Naive Bayes Classifiers?

Naive Bayes classifiers are a collection of classification algorithms based on [Bayes’ Theorem](https://www.geeksforgeeks.org/bayes-theorem/). It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. To start with, let us consider a dataset.

One of the most simple and effective classification algorithms, the Naïve Bayes classifier aids in the rapid development of machine learning models with rapid prediction capabilities.

Naïve Bayes algorithm is used for classification problems. It is highly used in text classification. In text classification tasks, data contains high dimension (as each word represent one feature in the data). It is used in spam filtering, sentiment detection, rating classification etc. The advantage of using naïve Bayes is its speed. It is fast and making prediction is easy with high dimension of data.

This model predicts the probability of an instance belongs to a class with a given set of feature value. It is a probabilistic classifier. It is because it assumes that one feature in the model is independent of existence of another feature. In other words, each feature contributes to the predictions with no relation between each other. In real world, this condition satisfies rarely. It uses Bayes theorem in the algorithm for training and prediction

## Advantages of Naive Bayes Classifier

* Easy to implement and computationally efficient.
* Effective in cases with a large number of features.
* Performs well even with limited training data.
* It performs well in the presence of categorical features.
* For numerical features data is assumed to come from normal distributions

## Disadvantages of Naive Bayes Classifier

* Assumes that features are independent, which may not always hold in real-world data.
* Can be influenced by irrelevant attributes.
* May assign zero probability to unseen events, leading to poor generalization

## Applications of Naive Bayes Classifier

* **Spam Email Filtering**: Classifies emails as spam or non-spam based on features.
* **Text Classification**: Used in sentiment analysis, document categorization, and topic classification.
* **Medical Diagnosis:** Helps in predicting the likelihood of a disease based on symptoms.
* **Credit Scoring:** Evaluates creditworthiness of individuals for loan approval.
* **Weather Prediction**: Classifies weather conditions based on various factors.

# Support Vector Machine (SVM) Algorithm

# Support Vector Machine (SVM) is a [supervised machine learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/) algorithm used for both classification and regression. Though we say regression problems as well it’s best suited for classification. The main objective of the SVM algorithm is to find the optimal [hyperplane](https://www.geeksforgeeks.org/separating-hyperplanes-in-svm/) in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

### Types of Support Vector Machine

Based on the nature of the decision boundary, Support Vector Machines (SVM) can be divided into two main parts:

* **Linear SVM:**Linear SVMs use a linear decision boundary to separate the data points of different classes. When the data can be precisely linearly separated, linear SVMs are very suitable. This means that a single straight line (in 2D) or a hyperplane (in higher dimensions) can entirely divide the data points into their respective classes. A hyperplane that maximizes the margin between the classes is the decision boundary.
* **Non-Linear SVM:** Non-Linear SVM can be used to classify data when it cannot be separated into two classes by a straight line (in the case of 2D). By using kernel functions, nonlinear SVMs can handle nonlinearly separable data. The original input data is transformed by these kernel functions into a higher-dimensional feature space, where the data points can be linearly separated. A linear SVM is used to locate a nonlinear decision boundary in this modified space.

### **Advantages of SVM**

* Effective in high-dimensional cases.
* Its memory is efficient as it uses a subset of training points in the decision function called support vectors.
* Different kernel functions can be specified for the decision functions and its possible to specify custom kernels.

# Understanding the Confusion Matrix in Machine Learning

## What is a Confusion Matrix?

A **confusion matrix** is a matrix that summarizes the performance of a machine learning model on a set of test data. It is a means of displaying the number of accurate and inaccurate instances based on the model’s predictions. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance.

The matrix displays the number of instances produced by the model on the test data.

* **True Positive (TP):** The model correctly predicted a positive outcome (the actual outcome was positive).
* **True Negative (TN):** The model correctly predicted a negative outcome (the actual outcome was negative).
* **False Positive (FP):** The model incorrectly predicted a positive outcome (the actual outcome was negative). Also known as a Type I error.
* **False Negative (FN):** The model incorrectly predicted a negative outcome (the actual outcome was positive). Also known as a Type II error.

# Precision and Recall in Machine Learning

## **What is Precision?**

Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples (either correctly or incorrectly).

1. Precision = True Positive/True Positive + False Positive
2. Precision = TP/TP+FP

* TP- True Positive
* FP- False Positive
* The precision of a machine learning model will be low when the value of;

1. TP+FP (denominator) > TP (Numerator)

* The precision of the machine learning model will be high when Value of;

1. TP (Numerator) > TP+FP (denominator)

## **What is Recall?**

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The **recall measures the model's ability to detect positive samples**. The higher the recall, the more positive samples detected.

1. Recall = True Positive/True Positive + False Negative
2. Recall = TP/TP+FN

* TP- True Positive
* FN- False Negative
* Recall of a machine learning model will be low when the value of;  
  TP+FN (denominator) > TP (Numerator)
* Recall of machine learning model will be high when Value of;  
  TP (Numerator) > TP+FN (denominator)

# F1 Score in Machine Learning

## What is an F1 score?

The F1 score is calculated as the harmonic mean of precision and recall. A[harmonic mean](https://www.geeksforgeeks.org/harmonic-mean/) is a type of average calculated by summing the reciprocal of each value in a data set and then dividing the number of values in the dataset by that sum. The value of the F1 score lies between 0 to 1 with 1 being a better

**1. Precision:** Precision represents the accuracy of positive predictions. It calculates how often the model predicts correctly the positive values. It is the number of true positive predictions divided by the total number of positive predictions (true positives + false positives).

It could be there are 10 positive cases and 5 negative cases. The model can identify 5 positive cases. But out of these 5 identified cases, 4 positive cases only 4 are positive and 1 is negative. Thus precision becomes 80% (4/5)

**2. Recall (Sensitivity or True Positive Rate)**: Recall represents how well a model can identify actual positive cases. It is the number of true positive predictions divided by the total number of actual positive instances (true positives + false negatives). It measures the ability of the model to capture all positive instances.

Taking the above example though the accuracy of predicting the positive case is very high(precision 80%) the recall will be very poor as out of the actual 10 positive case model was able to identify only 4 positive cases. Thus recall comes to (4/10) = 40%

There is often an inverse relationship between precision and recall. There could be cases depending on the domain where we would want either precision or recall to be an important metric. However, generally, we would want a model that can perform better on both. This is where the F1 metric comes into the picture.

F1 score combines precision and recall into a single metric

# AUC ROC Curve in Machine Learning

## What is the AUC-ROC curve?

The AUC-ROC curve, or Area Under the Receiver Operating Characteristic curve, is a graphical representation of the performance of a binary classification model at various classification thresholds. It is commonly used in machine learning to assess the ability of a model to distinguish between two classes, typically the positive class (e.g., presence of a disease) and the negative class (e.g., absence of a disease).

Let’s first understand the meaning of the two terms **ROC**and**AUC**.

* **ROC**: Receiver Operating Characteristics
* **AUC**: Area Under Curve

### Area Under Curve**(AUC) Curve:**

AUC stands for the Area Under the Curve, and the AUC curve represents the area under the ROC curve. It measures the overall performance of the binary classification model. As both TPR and FPR range between 0 to 1, So, the area will always lie between 0 and 1, and A greater value of AUC denotes better model performance. Our main goal is to maximize this area in order to have the highest TPR and lowest FPR at the given threshold. The AUC measures the probability that the model will assign a randomly chosen positive instance a higher predicted probability compared to a randomly chosen negative instance

# Nonlinear Regression

## Non-linear regression in Machine Learning

Nonlinear regression refers to a broader category of regression models where the relationship between the dependent variable and the independent variables is not assumed to be linear. If the underlying pattern in the data exhibits a curve, whether it’s exponential growth, decay, logarithmic, or any other non-linear form, fitting a nonlinear regression model can provide a more accurate representation of the relationship. This is because in linear regression it is pre-assumed that the data is linear.

A nonlinear regression model can be expressed as:

Where,

* : Regression function
* **X:** This is the vector of independent variables, which are used to predict the dependent variable.
* :The vector of parameters that the model aims to estimate. These parameters determine the shape and characteristics of the regression function.
* **:** error term

Many different regressions exist and can be used to fit whatever the dataset looks like such as quadratic, cubic regression, and so on to infinite degrees according to our requirement.

## Types of Non-Linear Regression

There are two main types of Non Linear regression in Machine Learning:

1. **Parametric non-linear regression** assumes that the relationship between the dependent and independent variables can be modeled using a specific mathematical function. For example, the relationship between the population of a country and time can be modeled using an exponential function. Some common parametric non-linear regression models include:Polynomial regression, Logistic regression, Exponential regression, Power regression etc.
2. **Non-parametric non-linear regression** does not assume that the relationship between the dependent and independent variables can be modeled using a specific mathematical function. Instead, it uses machine learning algorithms to learn the relationship from the data. Some common non-parametric non-linear regression algorithms include: Kernel smoothing, Local polynomial regression, Nearest neighbor regression etc.

## Evaluating Non-Linear Regression Models

Evaluating the performance of a nonlinear regression model is crucial to ensure it accurately represents the underlying relationship between the independent and dependent variables.

There are a number of different metrics that can be used to evaluate non-linear regression models, but some common metrics are:

1. **R-squared**– R-squared (Coefficient of Determination) measures the proportion of variance in the dependent variable that is explained by the independent variables in the model. It ranges from 0 to 1, where 0 indicates no explanation of variance and 1 indicates perfect explanation. A higher R-squared value suggests a better model fit.
2. **Adjusted R-squared** – Adjusted R-squared is a modified version of R-squared that accounts for the number of independent variables in the model. It penalizes models with more variables, making it a more appropriate measure of goodness of fit when comparing models with different numbers of independent variables. A higher adjusted R-squared value indicates a better model fit.
3. **Root Mean Squared Error (RMSE)** – Root Mean Squared Error (RMSE) is the square root of MSE, providing a more intuitive measure of the average error in predictions. It represents the average distance between the predicted and actual values of the dependent variable, scaled to the same units as the dependent variable. A lower RMSE signifies a better model fit.

## Linear VS Non-Linear Regression

| **Feature** | **Linear Regression** | **Non Linear Regression** |
| --- | --- | --- |
| Relationship between variables | Assumes a linear relationship between the independent and dependent variables | Allows for non-linear relationships between the independent and dependent variables |
| Model complexity | Simpler model with fewer parameters | More complex model with more parameters |
| Interpretability | Highly interpretable due to the linear relationship | Less interpretable due to the non-linear relationship |
| Overfitting susceptibility | More susceptible to overfitting due to its simplicity | Less susceptible to overfitting due to its ability to capture complex relationships |
| Flexibility | Requires large datasets to accurately estimate the linear relationship | Can work with smaller datasets due to its flexibility |
| Applications | Suitable for predicting continuous target variables when the relationship is linear | Suitable for predicting continuous target variables when the relationship is non-linear |
| Examples | Predicting house prices based on size and location | Predicting customer churn based on behavioral patterns |

## **Advantages** & **Disadvantages of Non-Linear Regression**

### **Advantages of Non-Linear Regression**

1. Non-linear regression can model relationships that are not linear in nature.
2. Non-linear regression can be used to make predictions about the dependent variable based on the values of the independent variables.
3. Non-linear regression can be used to identify the factors that influence the dependent variable.

### **Disadvantages of Non-Linear Regression**

1. Non-linear regression models can be more complex to implement than linear regression.
2. Non-linear regression models can be more sensitive to outliers than linear regression models.
3. Non-linear regression models can be more computationally expensive to train than linear regression models.