1. What is the concept of supervised learning? What is the significance of the name?

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.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.

In the real-world, supervised learning can be used for **Risk Assessment, Image classification, Fraud Detection, spam filtering**, etc.

How Supervised Learning Works?

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

The working of Supervised learning can be easily understood by the below example and diagram:



Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

* If the given shape has four sides, and all the sides are equal, then it will be labelled as a **Square**.
* If the given shape has three sides, then it will be labelled as a **triangle**.
* If the given shape has six equal sides then it will be labelled as **hexagon**.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape.

The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

Steps Involved in Supervised Learning:

* First Determine the type of training dataset
* Collect/Gather the labelled training data.
* Split the training dataset into training **dataset, test dataset, and validation dataset**.
* Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
* Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
* Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
* Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems:



**1. Regression**

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

* Linear Regression
* Regression Trees
* Non-Linear Regression
* Bayesian Linear Regression
* Polynomial Regression

**2. Classification**

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

Spam Filtering,

* Random Forest
* Decision Trees
* Logistic Regression
* Support vector Machines

Note: We will discuss these algorithms in detail in later chapters.

Advantages of Supervised learning:

* With the help of supervised learning, the model can predict the output on the basis of prior experiences.
* In supervised learning, we can have an exact idea about the classes of objects.
* Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

Disadvantages of supervised learning:

* Supervised learning models are not suitable for handling the complex tasks.
* Supervised learning cannot predict the correct output if the test data is different from the training dataset.
* Training required lots of computation times.
* In supervised learning, we need enough knowledge about the classes of object.

1. In the hospital sector, offer an example of supervised learning.

The growing number of applications of machine learning in healthcare allows the health care industries to manage their data and enhance their services effectively. Let’s look at a couple of applications of machine learning in the healthcare industry.

### 1. Managing Medical Data

Machine learning is actually advancing the health care industry by implementing cognitive technology in order to unwind a huge amount of medical records and also in order to perform any power diagnosis. Machine learning helps predict the intent of a user. Implementing machine learning in an organization’s workflow can develop a personalized user experience that allows the company to make better decisions and better actions that enhance the customer’s experience which benefits the organization. Therefore, machine learning helps to store, collect and reform data.

### 2. Helps in Medical Diagnosis

According to the Global Market Insights, Medical imaging and diagnosis powered by AI should witness more than 40% growth to surpass $2.5 billion by 2024. With the help of machine learning and deep learning models, AI is actually revolutionizing the image diagnosis field in medicine. The one major application of AI in medical diagnosis is MRI scans. AI has taken over the complex analysis of MRI scans and it has made it a much simpler process.

### 3. Detecting Diseases at an Earlier Stage

Machine learning played a very important role in the early predictions of medical conditions such as heart attacks and diabetes. There are many AI-based wearables that are being developed to monitor the health of a person and display any warnings when the devices observe something unusual or unlikely. For eg:- Fitbit and Apple watch. These devices monitor a person’s heart rate, sleep cycle, breathing rate, activity level, blood pressure, so on. It keeps the record of these measures 24×7.

### 4. Machine Learning in Medical Assistance

As an engine for medical assistants has grown the development of artificial intelligence-based virtual nurses has increased according to a recent survey, virtual Nursing Assistants corresponds to a maximum of 20 billion u.s dollars by 2027. A virtual nurse helps to monitor patients’ conditions and follow up with treatment between the doctor visits.

### 5. Machine Learning in Decision Making

AI has played a very important role in decision-making not only in the field of health care, but AI has also improved businesses by studying customer needs and evaluating any potential risk that a business might face. A powerful use case of artificial intelligence in decision-making is the use of surgical robots that can minimize errors and any variations and will eventually help in increasing the efficiency of your surgeons. They help implement complex surgeries with better flexibility and control than any other approach.

### 6. Personalized Medicine

The predictive analysis of Machine learning can help users to get personalized treatment. Generally, nurses are bound to choose from a specific set of diagnoses or predict the risk to the patient using a fixed formula based on the history and available genetic information. Whereas, machine learning in medicine predicts the data of the patient by analyzing the medical history to generate multiple treatment options. Due to these treatments being based on the user’s data they’re more likely to suit the patient and are more personalized.

### 7. Helps Analyze the Errors in Prescriptions

Machine learning can detect and analyze the errors in the prescriptions too. The intelligence can inspect the patient’s health records with the given prescriptions to find and correct the possible errors in the medication.

**Conclusion:**In health care industries, the intelligence by various data science models, without any human intelligence, can provide accurate and efficient results with little to no time. The machine learning models help health care sectors in numerous ways, by establishing accurate and efficient suggestions or by reducing the manual tasks of healthcare professionals, hence allowing them to focus on the research area and enhance their performance in urgent cases.

1. Give three supervised learning examples.

### Email Filtering

Supervised learning is commonly used in email filtering to classify incoming emails as spam or legitimate. A machine learning algorithm is trained using a labeled dataset containing examples of both spam and legitimate emails. The algorithm then extracts relevant information from each email, such as the sender’s information, the subject, the message body, and so on. It learns from the labeled dataset to identify patterns and relationships between these features and their corresponding labels (spam or legitimate). Once trained, the algorithm can use the extracted features to predict the label of new, unseen emails. If an email is predicted to be spam, it can be automatically filtered into a spam folder, saving the user’s inbox space.

### Credit Scoring

In credit scoring, supervised learning is used to predict the creditworthiness of loan applicants. A labeled dataset containing examples of past loan applicants and their credit history, income, employment status, and other relevant factors is used to train a machine learning algorithm. The algorithm learns to recognize patterns and relationships between these features and their corresponding labels, such as whether or not the loan was repaid. Once trained, the algorithm can predict loan repayment likelihood for new loan applicants based on their input features.

### Voice Recognition

Supervised learning is utilized in voice recognition to help virtual assistants and other applications recognize and understand spoken commands. A labeled dataset of spoken words and phrases with corresponding text transcripts is used to train a machine learning algorithm in such scenarios. The algorithm learns to recognize relationships between spoken word audio features such as pitch, amplitude, and frequency and their textual representations from the labeled dataset. Following the training phase, the algorithm can begin analyzing new audio inputs and attempting to transcribe them into text form. This allows virtual assistants to understand and respond to spoken commands like managing reminders, playing music, or controlling smart home devices.

1. In supervised learning, what are classification and regression?

# Classification vs Regression in Machine Learning

**Last Updated :**06 Nov, 2023

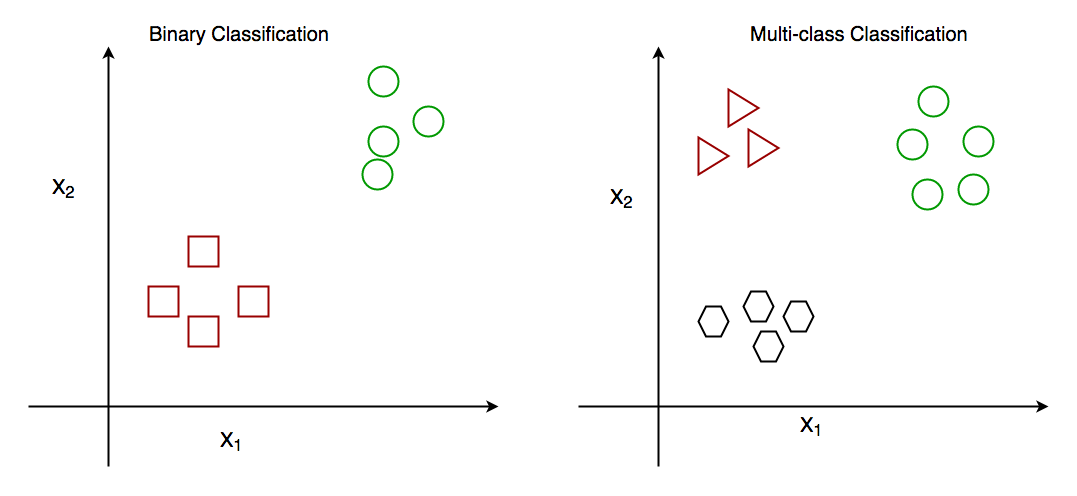
Classification and Regression are two major prediction problems that are usually dealt with in [Data Mining](https://www.geeksforgeeks.org/data-mining/) and [Machine Learning](https://www.geeksforgeeks.org/machine-learning/). We are going to deal with both Classification and Regression and we will also see differences between them in this article.

## Classification Algorithms

[**Classification**](https://www.geeksforgeeks.org/getting-started-with-classification/) is the process of finding or discovering a model or function that helps in separating the data into multiple categorical classes i.e. discrete values. In classification, data is categorized under different labels according to some parameters given in the input and then the labels are predicted for the data.

* In a classification task, we are supposed to predict discrete target variables(class labels) using independent features.
* In the classification task, we are supposed to find a [decision boundary](https://www.geeksforgeeks.org/ml-decision-function/) that can separate the different classes in the target variable.

The derived mapping function could be demonstrated in the form of “IF-THEN” rules. The classification process deals with problems where the data can be divided into binary or multiple discrete labels. Let’s take an example, suppose we want to predict the possibility of the winning of a match by Team A on the basis of some parameters recorded earlier. Then there would be two labels Yes and No.



*Binary Classification and Multiclass Classification*

### Types of Classification Algorithms

There are different types of State of the art classification algorithms that have been developed over time to give the best results for classification tasks by employing techniques like [bagging](https://www.geeksforgeeks.org/ml-bagging-classifier/) and [boosting](https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/).

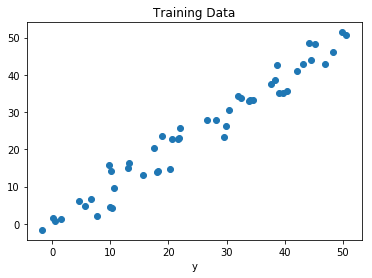
* [Decision Tree](https://www.geeksforgeeks.org/decision-tree/)
* [Random Forest Classifier](https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/)
* [K – Nearest Neighbors](https://www.geeksforgeeks.org/k-nearest-neighbours/)
* [Support Vector Machine](https://www.geeksforgeeks.org/support-vector-machine-algorithm/)

## Regression Algorithms

[**Regression**](https://www.geeksforgeeks.org/ml-linear-regression/) is the process of finding a model or function for distinguishing the data into continuous real values instead of using classes or discrete values. It can also identify the distribution movement depending on the historical data. Because a regression predictive model predicts a quantity, therefore, the skill of the model must be reported as an error in those predictions.

* In a regression task, we are supposed to predict a continuous target variable using independent features.
* In the regression tasks, we are faced with generally two types of problems linear and non-linear regression.

Let’s take a similar example in regression also, where we are finding the possibility of rain in some particular regions with the help of some parameters recorded earlier. Then there is a probability associated with the rain.



*Regression of Day vs Rainfall (in mm)*

### Types of Regression Algorithms

There are different types of State of the art regression algorithms that have been developed over time to give the best results for regression tasks by employing techniques like bagging and boosting.

* [Lasso Regression](https://www.geeksforgeeks.org/implementation-of-lasso-regression-from-scratch-using-python/)
* [Ridge Regression](https://www.geeksforgeeks.org/implementation-of-ridge-regression-from-scratch-using-python/)
* [XGBoost Regressor](https://www.geeksforgeeks.org/xgboost-for-regression/)
* [LGBM Regressor](https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/)

## Comparison between Classification and Regression

| **Classification** | **Regression** |
| --- | --- |
| In this problem statement, the target variables are discrete. | In this problem statement, the target variables are continuous. |
| Problems like [Spam Email Classification](https://www.geeksforgeeks.org/detecting-spam-emails-using-tensorflow-in-python/), [Disease prediction](https://www.geeksforgeeks.org/disease-prediction-using-machine-learning/) like problems are solved using Classification Algorithms. | Problems like [House Price Prediction](https://www.geeksforgeeks.org/house-price-prediction-using-machine-learning-in-python/), [Rainfall Prediction](https://www.geeksforgeeks.org/ml-rainfall-prediction-using-linear-regression/) like problems are solved using regression Algorithms. |
| In this algorithm, we try to find the best possible decision boundary which can separate the two classes with the maximum possible separation. | In this algorithm, we try to find the best-fit line which can represent the overall trend in the data. |
| [Evaluation metrics](https://www.geeksforgeeks.org/metrics-for-machine-learning-model/) like Precision, Recall, and F1-Score are used here to evaluate the performance of the classification algorithms. | Evaluation metrics like [Mean Squared Error,](https://www.geeksforgeeks.org/python-mean-squared-error/) [R2-Score](https://www.geeksforgeeks.org/ml-r-squared-in-regression-analysis/), and  [MAPE](https://www.geeksforgeeks.org/how-to-calculate-mape-in-python/) are used here to evaluate the performance of the regression algorithms. |
| Here we face the problems like [binary Classification](https://www.geeksforgeeks.org/getting-started-with-classification/) or [Multi-Class Classification](https://www.geeksforgeeks.org/multiclass-classification-using-scikit-learn/) problems. | Here we face the problems like [Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/) models as well as non-linear models. |
| Input Data are Independent variables and categorical dependent variable. | Input Data are Independent variables and continuous dependent variable. |
| The classification algorithm’s task mapping the input value of x with the discrete output variable of y. | The regression algorithm’s task is mapping input value (x) with continuous output variable (y). |
| Output is Categorical labels. | Output is Continuous numerical values. |
| Objective is to  Predict categorical/class labels. | Objective is to Predicting continuous numerical values. |
| Example use cases are Spam detection, image recognition, sentiment analysis | Example use cases are Stock price prediction, house price prediction, demand forecasting. |
| **Examples of classification algorithms are:**  Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (K-NN), Naive Bayes, Neural Networks, K-Means Clustering, Multi-layer Perceptron (MLP), etc. | **Examples of regression algorithms are:**  Linear Regression, Polynomial Regression, Ridge Regression, Lasso Regression, Support Vector Regression (SVR), Decision Trees for Regression, Random Forest Regression, K-Nearest Neighbors (K-NN) Regression, Neural Networks for Regression, etc. |

1. Give some popular classification algorithms as examples.

Classification algorithms can be used in different places. Below are some popular use cases of Classification Algorithms:

* Email Spam Detection
* Speech Recognition
* Identifications of Cancer tumor cells.
* Drugs Classification
* Biometric Identification, etc

1. Briefly describe the SVM model.

# Support Vector Machine (SVM) Algorithm

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature.

## Support Vector Machine

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.

Let’s consider two independent variables x1, x2, and one dependent variable which is either a blue circle or a red circle.



*Linearly Separable Data points*

From the figure above it’s very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregate our data points or do a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points?

### How does SVM work?

One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.



*Multiple hyperplanes separate the data from two classes*

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**. So from the above figure, we choose L2. Let’s consider a scenario like shown below



*Selecting hyperplane for data with outlier*

Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It’s simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.



*Hyperplane which is the most optimized one*

So in this type of data point what SVM does is, finds the maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these types of cases are called **soft margins**. When there is a soft margin to the data set, the SVM tries to minimize (1/margin+∧(∑penalty)). Hinge loss is a commonly used penalty. If no violations no hinge loss.If violations hinge loss proportional to the distance of violation.

Till now, we were talking about linearly separable data(the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?



*Original 1D dataset for classification*

Say, 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



*Mapping 1D data to 2D to become able to separate the two classes*

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 referred to as a kernel.

### Support Vector Machine Terminology

1. **Hyperplane:**Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space. In the case of linear classifications, it will be a linear equation i.e. wx+b = 0.
2. **Support Vectors:**Support vectors are the closest data points to the hyperplane, which makes a critical role in deciding the hyperplane and margin.
3. **Margin**: Margin is the distance between the support vector and hyperplane. The main objective of the support vector machine algorithm is to maximize the margin.  The wider margin indicates better classification performance.
4. **Kernel**: Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces, so, that the hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.
5. **Hard Margin:** The maximum-margin hyperplane or the hard margin hyperplane is a hyperplane that properly separates the data points of different categories without any misclassifications.
6. **Soft Margin:**When the data is not perfectly separable or contains outliers, SVM permits a soft margin technique. Each data point has a slack variable introduced by the soft-margin SVM formulation, which softens the strict margin requirement and permits certain misclassifications or violations. It discovers a compromise between increasing the margin and reducing violations.
7. **C:**Margin maximisation and misclassification fines are balanced by the regularisation parameter C in SVM. The penalty for going over the margin or misclassifying data items is decided by it. A stricter penalty is imposed with a greater value of C, which results in a smaller margin and perhaps fewer misclassifications.
8. **Hinge Loss:** A typical loss function in SVMs is hinge loss. It punishes incorrect classifications or margin violations. The objective function in SVM is frequently formed by combining it with the regularisation term.
9. **Dual Problem:** A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The dual formulation enables the use of kernel tricks and more effective computing.

### Mathematical intuition of Support Vector Machine

Consider a binary classification problem with two classes, labeled as +1 and -1. We have a training dataset consisting of input feature vectors X and their corresponding class labels Y.

The equation for the linear hyperplane can be written as:

𝑤𝑇𝑥+𝑏=0*wTx*+*b*=0

The vector W represents the normal vector to the hyperplane. i.e the direction perpendicular to the hyperplane. The parameter **b** in the equation represents the offset or distance of the hyperplane from the origin along the normal vector **w**.

The distance between a data point x\_i and the decision boundary can be calculated as:

𝑑𝑖=𝑤𝑇𝑥𝑖+𝑏∣∣𝑤∣∣*di*​=∣∣*w*∣∣*wTxi*​+*b*​

where ||w|| represents the Euclidean norm of the weight vector w. Euclidean norm of the normal vector W

For Linear SVM classifier :

𝑦^={1: 𝑤𝑇𝑥+𝑏≥00:  𝑤𝑇𝑥+𝑏 <0*y*^​={10​: *wTx*+*b*≥0:  *wTx*+*b* <0​

#### ****Optimization:****

* **For Hard margin linear SVM classifier:**

𝑚𝑖𝑛𝑖𝑚𝑖𝑧𝑒𝑤,𝑏12𝑤𝑇𝑤=𝑚𝑖𝑛𝑖𝑚𝑖𝑧𝑒𝑊,𝑏12∥𝑤∥2subject to𝑦𝑖(𝑤𝑇𝑥𝑖+𝑏)≥1𝑓𝑜𝑟𝑖=1,2,3,⋯,𝑚*w*,*b*minimize​21​*wTw*=*W*,*b*minimize​21​∥*w*∥2subject to*yi*​(*wTxi*​+*b*)≥1*fori*=1,2,3,⋯,*m*

The target variable or label for the ith training instance is denoted by the symbol ti in this statement. And ti=-1 for negative occurrences (when yi= 0) and ti=1positive instances (when yi = 1) respectively. Because we require the decision boundary that satisfy the constraint:  𝑡(𝑤𝑇𝑥𝑖+𝑏)≥1*ti*​(*wTxi*​+*b*)≥1

* **For Soft margin linear SVM classifier:**

𝑚𝑖𝑛𝑖𝑚𝑖𝑧𝑒 𝑤,𝑏12𝑤𝑇𝑤+𝐶∑𝑖=1𝑚𝜁𝑖subject to 𝑦𝑖(𝑤𝑇𝑥𝑖+𝑏)≥ 1−𝜁𝑖𝑎𝑛𝑑𝜁𝑖≥0𝑓𝑜𝑟𝑖=1,2,3,⋯,𝑚*w*,*b*minimize ​21​*wTw*+*C*∑*i*=1*m*​*ζi*​subject to *yi*​(*wTxi*​+*b*)≥ 1−*ζi*​*andζi*​≥0*fori*=1,2,3,⋯,*m*

* **Dual Problem:** A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The optimal Lagrange multipliers α(i) that maximize the following dual objective function

𝑚𝑎𝑥𝑖𝑚𝑖𝑧𝑒𝛼:12∑𝑖→𝑚∑𝑗→𝑚𝛼𝑖𝛼𝑗𝑡𝑖𝑡𝑗(𝑥𝑖,𝑥𝑗)−∑𝑖→𝑚𝛼𝑖*α*maximize​:21​*i*→*m*∑​*j*→*m*∑​*αi*​*αj*​*ti*​*tj*​*K*(*xi*​,*xj*​)−*i*→*m*∑​*αi*​

where,

* αi is the Lagrange multiplier associated with the ith training sample.
* K(xi, xj) is the kernel function that computes the similarity between two samples xi and xj. It allows SVM to handle nonlinear classification problems by implicitly mapping the samples into a higher-dimensional feature space.
* The term ∑αi represents the sum of all Lagrange multipliers.

The SVM decision boundary can be described in terms of these optimal Lagrange multipliers and the support vectors once the dual issue has been solved and the optimal Lagrange multipliers have been discovered. The training samples that have i > 0 are the support vectors, while the decision boundary is supplied by:

𝑤=∑𝑖→𝑚𝛼𝑖𝑡𝑖(𝑥𝑖,𝑥)+𝑏𝑡𝑖(𝑤𝑇𝑥𝑖−𝑏)=1⟺𝑏=𝑤𝑇𝑥𝑖−𝑡𝑖*w*=*i*→*m*∑​*αi*​*ti*​*K*(*xi*​,*x*)+*bti*​(*wTxi*​−*b*)=1⟺*b*=*wTxi*​−*ti*​

### 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.

### Popular kernel functions in SVM

The SVM kernel is a function that takes low-dimensional input space and transforms it into higher-dimensional space, ie it converts nonseparable problems to separable problems. It is mostly useful in non-linear separation problems. Simply put the kernel, does some extremely complex data transformations and then finds out the process to separate the data based on the labels or outputs defined.

Linear : 𝐾(𝑤,𝑏)=𝑤𝑇𝑥+𝑏Polynomial : 𝐾(𝑤,𝑥)=(𝛾𝑤𝑇𝑥+𝑏)𝑁Gaussian RBF: 𝐾(𝑤,𝑥)=exp⁡(−𝛾∣∣𝑥𝑖−𝑥𝑗∣∣𝑛Sigmoid :𝐾(𝑥𝑖,𝑥𝑗)=tanh⁡(𝛼𝑥𝑖𝑇𝑥𝑗+𝑏)Linear : *K*(*w*,*b*)Polynomial : *K*(*w*,*x*)Gaussian RBF: *K*(*w*,*x*)Sigmoid :*K*(*xi*​,*xj*​)​=*wTx*+*b*=(*γwTx*+*b*)*N*=exp(−*γ*∣∣*xi*​−*xj*​∣∣*n*=tanh(*αxiT*​*xj*​+*b*)​

### ****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.

### ****SVM implementation in Python****

Predict if cancer is Benign or malignant. Using historical data about patients diagnosed with cancer enables doctors to differentiate malignant cases and benign ones are given independent attributes.

#### Steps

* Load the breast cancer dataset from sklearn.datasets
* Separate input features and target variables.
* Build and train the SVM classifiers using RBF kernel.
* Plot the scatter plot of the input features.
* Plot the decision boundary.
* Plot the decision boundary

1. In SVM, what is the cost of misclassification?

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart, or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of less expensive errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select **Use misclassification costs** and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying A as B to be 2.0, the cost of misclassifying B as A will still have the default value of 1.0 unless you explicitly change it as well.

1. In the SVM model, define Support Vectors.

Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



**Example:** SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram

1. In the SVM model, define the kernel.

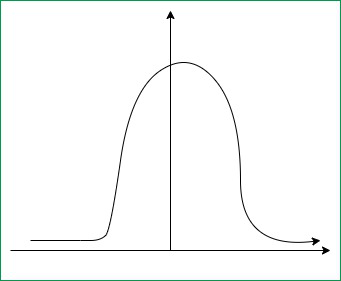
**Major Kernel Functions in Support Vector Machine (SVM)**

**Kernel Function** is a method used to take data as input and transform it into the required form of processing data. “Kernel” is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces. Basically, It returns the inner product between two points in a standard feature dimension.   
**Standard Kernel Function Equation :**

**Major Kernel Functions :-**   
For Implementing Kernel Functions, first of all, we have to install the “scikit-learn” library using the command prompt terminal: 

pip install scikit-learn

* **Gaussian Kernel:**It is used to perform transformation when there is no prior knowledge about data.
* **Gaussian Kernel Radial Basis Function (RBF):**Same as above kernel function, adding radial basis method to improve the transformation.



***Gaussian Kernel Graph***

1. What are the factors that influence SVM's effectiveness?
2. **Hyperplane:**Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space. In the case of linear classifications, it will be a linear equation i.e. wx+b = 0.
3. **Support Vectors:**Support vectors are the closest data points to the hyperplane, which makes a critical role in deciding the hyperplane and margin.
4. **Margin**: Margin is the distance between the support vector and hyperplane. The main objective of the support vector machine algorithm is to maximize the margin.  The wider margin indicates better classification performance.
5. **Kernel**: Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces, so, that the hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.
6. **Hard Margin:** The maximum-margin hyperplane or the hard margin hyperplane is a hyperplane that properly separates the data points of different categories without any misclassifications.
7. **Soft Margin:**When the data is not perfectly separable or contains outliers, SVM permits a soft margin technique. Each data point has a slack variable introduced by the soft-margin SVM formulation, which softens the strict margin requirement and permits certain misclassifications or violations. It discovers a compromise between increasing the margin and reducing violations.
8. **C:**Margin maximisation and misclassification fines are balanced by the regularisation parameter C in SVM. The penalty for going over the margin or misclassifying data items is decided by it. A stricter penalty is imposed with a greater value of C, which results in a smaller margin and perhaps fewer misclassifications.
9. **Hinge Loss:** A typical loss function in SVMs is hinge loss. It punishes incorrect classifications or margin violations. The objective function in SVM is frequently formed by combining it with the regularisation term.
10. **Dual Problem:** A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The dual formulation enables the use of kernel tricks and more effective computing.
11. What are the benefits of using the SVM model?

* SVM works relatively well when there is a clear margin of separation between classes.
* SVM is more effective in high dimensional spaces.
* SVM is effective in cases where the number of dimensions is greater than the number of samples.
* SVM is relatively memory efficient

1. What are the drawbacks of using the SVM model?

* SVM algorithm is not suitable for large data sets.
* SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
* In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
* As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.

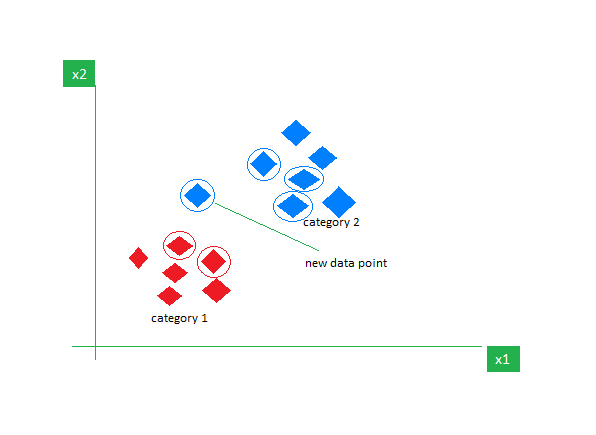
13. Notes should be written on

1. The kNN algorithm has a validation flaw.

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.

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



*KNN Algorithm working visualization*

Now, given another set of data points (also called testing data), allocate these points to a group by analyzing the training set. Note that the unclassified points are marked as ‘White’.

2. In the kNN algorithm, the k value is chosen.

The value of k is very crucial in the KNN algorithm to define the number of neighbors in the algorithm. The value of k in the k-nearest neighbors (k-NN) algorithm should be chosen based on the input data. If the input data has more outliers or noise, a higher value of k would be better. It is recommended to choose an odd value for k to avoid ties in classification. [Cross-validation](https://www.geeksforgeeks.org/cross-validation-machine-learning) methods can help in selecting the best k value for the given dataset.

3. A decision tree with inductive bias

**The inductive bias of decision trees is the assumption or preference that the algorithm makes when creating the tree structure.**

**One common inductive bias of trees is that they tend to favor shorter trees with simpler decision rules. This helps in reducing the complexity of the model and often leads to better generalization performance on unseen data.**

14.What are some of the benefits of the kNN algorithm?

Advantages of the KNN are the following. First, the algorithm is simple. Because it uses simple comparisons to find similar records in the training data, KNN is sometimes very effective at making good predictions.

Second, KNN can be effective at capturing complex interactions among variables without having to define a separable statistical model (no coefficients and weights).

Third, use of KNN requires no assumptions about the data. This is different than some other algorithms. For example, we learned that MLR works best when the data is normally distributed, when the relationships between predictors and outcome variables are approximately linear, and so forth. KNN can be effective without requiring such assumptions be met.

15. What are some of the kNN algorithm's drawbacks?

A disadvantage of the KNN algorithm is that it does not create a generalized separable model. There is no summary equations or trees that can be produced by the training process that can be quickly applied to new records. Instead, KNN simply uses the training data itself to perform prediction.

KNN provides no insight about the relative importance of each predictor.

Another significant disadvantage of KNN, is that the algorithm is computationally intensive. Computational effort of the algorithm increases greatly as more predictors, p, are considered and when the number of training records increase. The algorithm must compute the distance and find the nearest neighbors in all the training data for each prediction. This takes time and can be especially slow when there are a large number of training records that must be examined for each record to be predicted.

Including too many predictors will also make all instances in the training data far away from the record to be predicted. This is because expected distance to nearest neighbor increases with p. When the number of predictors is high, all records end up "far away" from each other because you are squaring and summing so many differences.

In situations where the number of records cannot be saved in active memory, the time costs of having to find many records in secondary storage such as hard drives can make the process slow because the mechanical process of spinning a hard drive to find the training records adds to the length of time required by the algorithm.

Because of this, KNN is best used when the number of training records is not great, when the number of predictors is not great, and when the training data can be saved in active computer memory.

16. Explain the decision tree algorithm in a few words.

**Decision Tree**

**Decision trees**are a popular and powerful tool used in various fields such as machine learning, data mining, and statistics. They provide a clear and intuitive way to make decisions based on data by modeling the relationships between different variables. This article is all about what decision trees are, how they work, their advantages and disadvantages, and their applications.

**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=1(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=1(∣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.

**Pruning**

To overcome **overfitting, pruning**techniques are used. Pruning reduces the size of the tree by removing nodes that provide little power in classifying instances. There are two main types of pruning:

* **Pre-pruning (Early Stopping)**: Stops the tree from growing once it meets certain criteria (e.g., maximum depth, minimum number of samples per leaf).
* **Post-pruning**: Removes branches from a fully grown tree that do not provide significant power.

**Applications of Decision Trees**

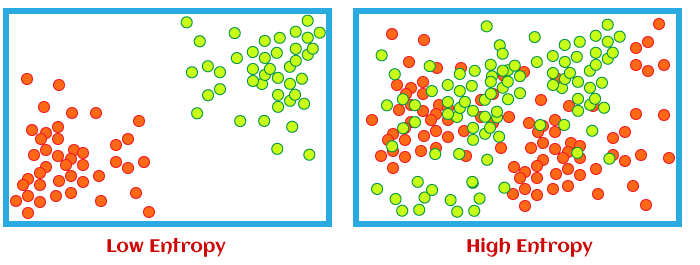
* **Business Decision Making**: Used in strategic planning and resource allocation.
* **Healthcare**: Assists in diagnosing diseases and suggesting treatment plans.
* **Finance**: Helps in credit scoring and risk assessment.
* **Marketing**: Used to segment customers and predict customer behavior.

17. What is the difference between a node and a leaf in a decision tree?

In a Decision tree, there are two nodes, which are the **Decision Node** and**Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches

18. What is a decision tree's entropy?

Entropy is the measurement of disorder or impurities in the information processed in machine learning. It determines how a decision tree chooses to split data.



We can understand the term entropy with any simple example: flipping a coin. When we flip a coin, then there can be two outcomes. However, it is difficult to conclude what would be the exact outcome while flipping a coin because there is no direct relation between flipping a coin and its outcomes. There is a 50% probability of both outcomes; then, in such scenarios, entropy would be high. This is the essence of entropy in machine learning.

19. In a decision tree, define knowledge gain.

* Decision Tree is a **Supervised learning technique**that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where**internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
* In a Decision tree, there are two nodes, which are the **Decision Node** and**Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
* The decisions or the test are performed on the basis of features of the given dataset.
* ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
* In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
* A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
* Below diagram explains the general structure of a decision tree:

20. Choose three advantages of the decision tree approach and write them down.

## . It’s Great for Making Decisions

If it wasn’t already obvious, we decided to make it the very first advantage to make it even more. The decision-tree algorithm is a constructive one that aids essential decision-making processes. For example, should your bank give a particular person a loan? Through this [algorithm](https://www.simplilearn.com/tutorials/data-structure-tutorial/what-is-an-algorithm), banks can use their website to sift through people who are eligible for loans from the people who may just be a lost deal. It does this by simply tracing the people who are eligible for the loan to a page that tells them to visit the bank. And the remaining individuals are led to other bank schemes that could help give them a lift.

## 2. It is an All-Inclusive Algorithm

Usually, data would have to be standardized before placing it into an algorithm. Although when it comes to decision trees, continuous and categorical variables can co-exist in the algorithm. This is because this machine learning algorithm does not rely on the inputs directly to predict an outcome. Instead, it depends upon the relation of the different inputs to predict what comes next, which is the outcome. For example, your click on the black shoes led you to a dozen ads from similar brands. As soon as you made the second click, it acknowledged how other customers had been taken down the same tree before, revealing the same search results to you. Somebody probably entered the data about previous customer results, and thanks to the decision tree, you were just led to it. If you had clicked on a different shoe, however, the tree would have taken you to another “branch”.

## 3. Simple to Understand for Coders

Decision trees can be visualized, which is exactly why coders have a much easier time dealing with this algorithm. They can quickly accumulate the data or information given by their clients, visualize it and deliver an algorithm. The functioning of this Decision tree algorithm is similar to the thought process of an actual human brain.

## 4. Missing Values Aren’t an Issue

Decision Trees are an absolute dream for coders because the algorithm can operate even with missing values. There are a few things that can take place. It could be an outcome of CART (classification and regression tree). CART is a predictive model which can predict the outcome regardless of the missing value based on other values, such as lean toward the most significant number of instances. It can also randomly assign you to a node (similar to a lead and endpoint in a branch). Either way, the algorithm does not stop working. Decision Trees will always configure beyond the missing values depending on the programming.

## 5. They are Inexpensive

Decision trees are surprisingly not that expensive because they make “greedy” decisions. At each stage of the fitting process, they multiply the sub-problem and find an optimal split with the data in the particular node and keep moving forward. They make choices based on only a linear scan of the data for that specific node. This is why, mostly, this method works relatively quickly, proving inexpensive computationally.

## 6. Less Coding for the Coder

Not only is it simpler to understand for coders, but the pre-processing steps of this algorithm require a lot less [coding](https://www.simplilearn.com/tutorials/programming-tutorial/coding-for-beginners) than most algorithms. This is because they don’t need analysis or even variables to keep them going. This is mainly because the technique involves looking at the inputs individually rather than the whole.

## Become a AI & Machine Learning Professional

* $267 billionExpected Global AI Market Value By 2027
* 40.2%The annual growth rate for the global AI market
* $15.7 trillionExpected Total Contribution Of AI To The Global Economy By 2030

#### Professional Certificate Course in Generative AI and Machine Learning

* + Program completion certificate from E&ICT Academy, IIT Kanpur
  + Curriculum delivered in live virtual classroom sessions by seasoned industry experts

11 Months months

[**View Program**](javascript:void(0))

#### Artificial Intelligence Engineer

* + Industry-recognized AI Engineer Master’s certificate from Simplilearn
  + Dedicated live sessions by faculty of industry experts

11 Months months

[**View Program**](javascript:void(0))

### Here's what learners are saying regarding our programs:

* 

#### Abhineet Srivastava

##### Senior Manager - Analytics & Reporting, AXA

The Machine Learning Course course was all worth it! The course content was comprehensive and updated. The journey from a Python-based approach to understanding Statistical concepts, Machine Learning, and other concepts was just incredible. Thanks to all the amazing trainers and co-learners for giving me such an enriched experience.

* Indrakala Nigam Beniwal

#### Indrakala Nigam Beniwal

##### Technical Consultant, Land Transport Authority (LTA) Singapore

I completed a Master's Program in Artificial Intelligence Engineer with flying colors from Simplilearn. Thanks to the course teachers and others associated with designing such a wonderful learning experience.

Not sure what you’re looking for?[View all Related Programs](https://www.simplilearn.com/ai-and-machine-learning?source=InpageBannerCategory)

## 7. Communicable to Important People

It should be clear by now how Decision Trees are visually appealing and easy to understand. This isn’t just an excellent thing for coders but also clients. Clients may not understand coding, analysis, inputs, and technical jargon, but everyone understands trees! It’s really easy to communicate to business stakeholders or important people what kind of application or code is under the process. In short, it’s easier to keep them in the loop!

## 8. It is Reliable

In a Decision Tree, it is effortless to trace each path to a conclusion. It ensures a comprehensive analysis of the consequences of each branch while also recognizing which nodes might need further analyzing. Therefore, it is easy to validate the algorithm using statistical tests. This makes Decision Trees an accountable model. And the ability to determine its accountability makes it reliable.

## 9. Can Handle Multiple Outputs

Decision Trees are one of the most important algorithms in machine learning because it is one that can generate many outputs. At the end of a Decision Tree, one can land upon several decisions. The capacity to handle the data of several outputs can direct you in a totally different direction than another 2 or 3 people who took the same Decision Tree route. That is however, given that the Decision Tree was initially fed with a bunch of data with multiple outputs.

21. Make a list of three flaws in the decision tree process.

**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.

22. Briefly describe the random forest model.

**Random Forest Algorithm in Machine Learning**

Machine learning, a fascinating blend of computer science and statistics, has witnessed incredible progress, with one standout algorithm being the **Random Forest**. **Random forests or Random Decision Trees** is a collaborative team of **decision trees** that work together to provide a single output. Originating in 2001 through Leo Breiman, Random Forest has become a cornerstone for machine learning enthusiasts. In this article, we will explore the fundamentals and implementation of **Random Forest Algorithm**.

**What is the Random Forest Algorithm?**

Random Forest algorithm is a powerful tree learning technique in [Machine Learning](https://www.geeksforgeeks.org/ml-machine-learning/). It works by creating a number of [Decision Trees](https://www.geeksforgeeks.org/decision-tree/) during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of [overfitting](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/) and improving overall prediction performance.

In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks) This collaborative decision-making process, supported by multiple trees with their insights, provides an example stable and precise results. Random forests are widely used for classification and regression functions, which are known for their ability to handle complex data, reduce overfitting, and provide reliable forecasts in different environments.

**What are Ensemble Learning models?**

Ensemble learning models work just like a group of diverse experts teaming up to make decisions – think of them as a bunch of friends with different strengths tackling a problem together. Picture it as a group of friends with different skills working on a project. Each friend excels in a particular area, and by combining their strengths, they create a more robust solution than any individual could achieve alone.

Similarly, in ensemble learning, different models, often of the same type or different types, team up to enhance predictive performance. It’s all about leveraging the collective wisdom of the group to overcome individual limitations and make more informed decisions in various machine learning tasks. Some popular ensemble models include- [XGBoost](https://www.geeksforgeeks.org/xgboost/), [AdaBoost](https://www.geeksforgeeks.org/differences-between-random-forest-and-adaboost/), [LightGBM](https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/), Random Forest, [Bagging](https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/), [Voting](https://www.geeksforgeeks.org/ml-voting-classifier-using-sklearn/) etc.

**What is Bagging and Boosting?**

Bagging is an ensemble learning model, where multiple week models are trained on different subsets of the training data. Each subset is sampled with replacement and prediction is made by averaging the prediction of the week models for regression problem and considering majority vote for classification problem.

Boosting trains multiple based models sequentially. In this method, each model tries to correct the errors made by the previous models. Each model is trained on a modified version of the dataset, the instances that were misclassified by the previous models are given more weight. The final prediction is made by weighted voting.

**Algorithm for Ransom Forest Work:**

1. Step 1: Select random K data points from the training set.
2. Step 2:Build the decision trees associated with the selected data points(Subsets).
3. Step 3:Choose the number N for decision trees that you want to build.
4. Step 4:Repeat Step 1 and 2.
5. Step 5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

**How Does Random Forest Work?**

The random Forest algorithm works in several steps which are discussed below–>

* **Ensemble of Decision Trees:** Random Forest leverages the power of [ensemble learning](https://www.geeksforgeeks.org/ensemble-methods-in-python/)by constructing an army of [Decision Trees](https://www.geeksforgeeks.org/decision-tree/). These trees are like individual experts, each specializing in a particular aspect of the data. Importantly, they operate independently, minimizing the risk of the model being overly influenced by the nuances of a single tree.
* **Random Feature Selection:** To ensure that each decision tree in the ensemble brings a unique perspective, Random Forest employs [random feature selection](https://www.geeksforgeeks.org/feature-selection-techniques-in-machine-learning/). During the training of each tree, a random subset of features is chosen. This randomness ensures that each tree focuses on different aspects of the data, fostering a diverse set of predictors within the ensemble.
* **Bootstrap Aggregating or Bagging:** The technique of bagging is a cornerstone of Random Forest’s training strategy which involves creating multiple bootstrap samples from the original dataset, allowing instances to be sampled with replacement. This results in different subsets of data for each decision tree, introducing variability in the training process and making the model more robust.
* **Decision Making and Voting:** When it comes to making predictions, each decision tree in the Random Forest casts its vote. For [classification tasks](https://www.geeksforgeeks.org/ml-classification-vs-regression/), the final prediction is determined by the [mode](https://www.geeksforgeeks.org/mode/) (most frequent prediction) across all the trees. In [regression tasks](https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/), the average of the individual tree predictions is taken. This internal voting mechanism ensures a balanced and collective decision-making process.

**Key Features of Random Forest**

Some of the Key Features of Random Forest are discussed below–>

1. **High Predictive Accuracy:** Imagine Random Forest as a team of decision-making wizards. Each wizard (decision tree) looks at a part of the problem, and together, they weave their insights into a powerful prediction tapestry. This teamwork often results in a more accurate model than what a single wizard could achieve.
2. **Resistance to Overfitting:** Random Forest is like a cool-headed mentor guiding its apprentices (decision trees). Instead of letting each apprentice memorize every detail of their training, it encourages a more well-rounded understanding. This approach helps prevent getting too caught up with the training data which makes the model less prone to overfitting.
3. **Large Datasets Handling:** Dealing with a mountain of data? Random Forest tackles it like a seasoned explorer with a team of helpers (decision trees). Each helper takes on a part of the dataset, ensuring that the expedition is not only thorough but also surprisingly quick.
4. **Variable Importance Assessment:** Think of Random Forest as a detective at a crime scene, figuring out which clues (features) matter the most. It assesses the importance of each clue in solving the case, helping you focus on the key elements that drive predictions.
5. **Built-in Cross-Validation:** Random Forest is like having a personal coach that keeps you in check. As it trains each decision tree, it also sets aside a secret group of cases (out-of-bag) for testing. This built-in validation ensures your model doesn’t just ace the training but also performs well on new challenges.
6. **Handling Missing Values:** Life is full of uncertainties, just like datasets with missing values. Random Forest is the friend who adapts to the situation, making predictions using the information available. It doesn’t get flustered by missing pieces; instead, it focuses on what it can confidently tell us.
7. **Parallelization for Speed:** Random Forest is your time-saving buddy. Picture each decision tree as a worker tackling a piece of a puzzle simultaneously. This parallel approach taps into the power of modern tech, making the whole process faster and more efficient for handling large-scale projects.