CLASSIFICATION

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observations into a number of classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc. Classes can be called as targets/labels or categories.

Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output.

In the classification algorithm, a discrete output function(y) is mapped to the input variable(x).

y=f(x), where y = categorical output

Classifications are discrete and do not imply order.

The algorithm which implements the classification on a dataset is known as a classifier. There are two types of Classifications:

Binary Classifier: If the classification problem has only two possible outcomes, then it is called a Binary Classifier.

Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.

Multi-class Classifier: If a classification problem has more than two outcomes, then it is called a Multi-class Classifier.

Example: Classifications of types of crops, Classification of types of music.

Types of ML Classification Algorithms:

Classification Algorithms can be further divided into the Mainly two category:

**Linear Models**

Logistic Regression

Support Vector Machines

**Non-linear Models**

K-Nearest Neighbors

Kernel SVM

Naïve Bayes

Decision Tree Classification

Random Forest Classification

Examples:

Given characteristics of individuals differentiate them who have suffered a heart attack from those who have not

Determine if a credit card purchase is fraudulent

Classify a car loan applicant as a good or a poor credit risk

Steps of classification process

(1) Train the model

using a training set

data objects whose class labels are known

(2) Test the model

on a test sample

whose class labels are known but not used for training the model

(3) Use the model for classification

classification model is constructed

– classifier is built describing a predetermined set of

data classes.

– Classification algorithm builds the classifier by

analyzing or learning from a training set made up of

database tuples and their associated labels.

--A tuple X is represented as n-dimensional attribute vector, X={x1,x2....xn}

--Each tuple X is assumed to belong to a predefined class as determined by another attribute.

The model is represented as classification rules, decision trees, or mathematical formulae

Predictive accuracy of the classifier is estimated.

if the accuracy is considered acceptable, the rule can be applied to the classification of new data tuples.

Decision Tree Induction

Decision Tree is a supervised learning method used in data mining for predictive problems in classification and regression methods.

It is a tree that helps us in decision-making purposes.

The decision tree creates classification or regression models as a tree structure of decisions and their possible consequences..

It separates a data set into smaller subsets, and at the same time, the decision tree is steadily developed. This process takes place recursively.

The final tree is a tree with the decision nodes and leaf nodes. A decision node has at least two branches. The leaf nodes show a classification or decision. We can't accomplish more split on leaf nodes-The uppermost decision node in a tree that relates to the best predictor called the root node.

There are two main types of decision trees: classification trees and regression trees. Classification trees are used to predict categorical variables, while regression trees are used to predict continuous variables.

Key factors:

Entropy:

Entropy refers to a common way to measure impurity. In the decision tree, it measures the randomness or impurity in data sets.

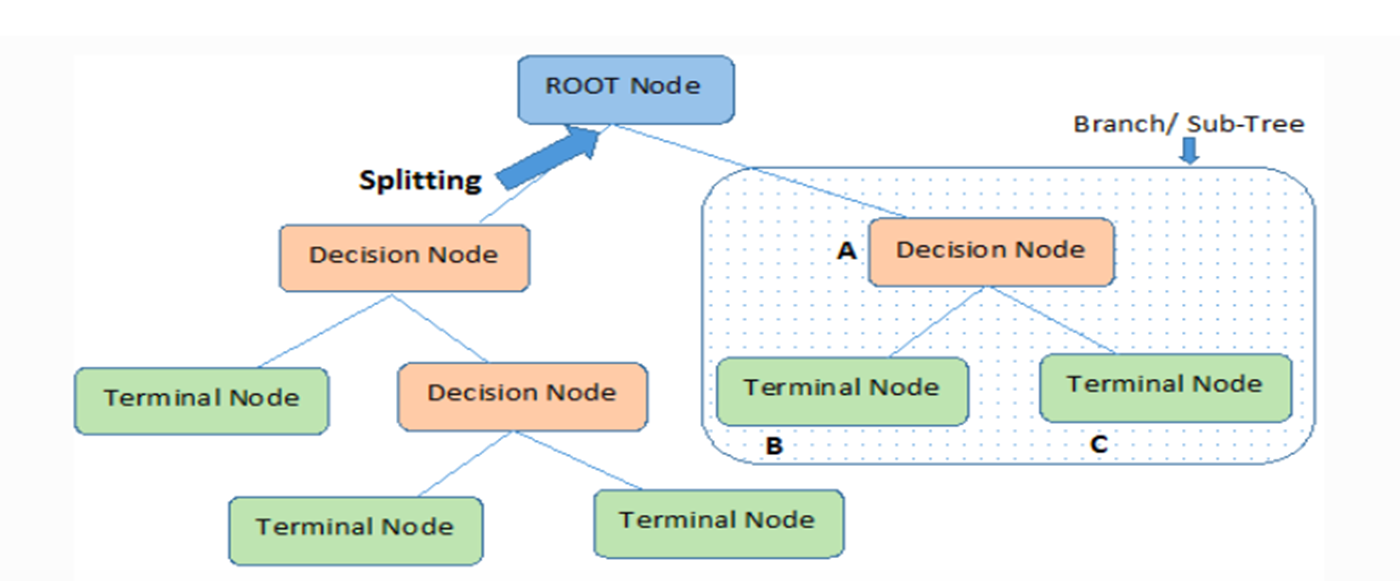
Information Gain:

Information Gain refers to the decline in entropy after the dataset is split. It is also called Entropy Reduction. Building a decision tree is all about discovering attributes that return the highest data gain.

Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node, with the leaf/terminal node providing the classification of the example.

Each node in the tree acts as a test case for some attribute, and each edge descending from the node corresponds to the possible answers to the test case.

This process is recursive in nature and is repeated for every subtree rooted at the new node.



Root Node: This attribute is used for dividing the data into two or more sets.

The feature attribute in this node is selected based on Attribute Selection Techniques. Root node represents the best attribute selected for classification.

Branch or Sub-Tree: A part of the entire decision tree is called a branch or sub-tree.

Splitting: Dividing a node into two or more sub-nodes based on if-else conditions.

Decision Node: After splitting the sub-nodes into further sub-nodes, then it is called the decision node.

Leaf or Terminal Node: This is the end of the decision tree where it cannot be split into further sub-nodes. Leaf node or terminal node which represents the classification or decision label.

Pruning: Removing a sub-node from the tree is called pruning.

The process of building a decision tree typically involves the following steps:

Attribute selection: The first step is to select the input variables (or attributes) that are most relevant to the prediction task. This is often done using a metric such as information gain or Gini index, which measures the impurity of the data at each attribute.

Tree construction: Once the attributes have been selected, the tree is constructed recursively by partitioning the data into subsets based on the values of each attribute. The algorithm chooses the attribute with the highest information gain or lowest Gini index to split the data at each level.

Pruning: After the tree has been constructed, it may be too complex and prone to overfitting. Pruning is a process of simplifying the tree by removing branches that do not improve its accuracy on a validation set.

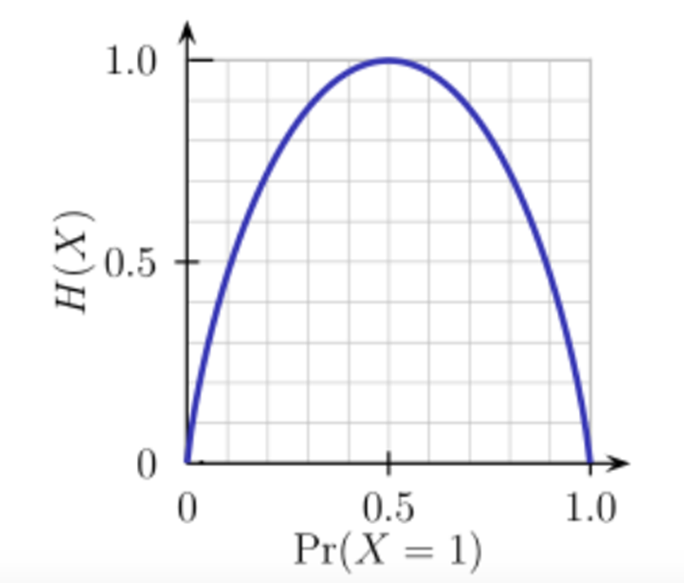
Tree evaluation: The final step is to evaluate the accuracy of the tree on a separate test set of data. This is done by calculating various performance metrics such as precision, recall, and F1-score.

ATTRIBUTE SELECTION MEASURES  
While using Information Gain as a criterion, we assume attributes to be categorical, and for the Gini index, attributes are assumed to be continuous.

Entropy

Entropy is a measure of the randomness or uncertainty in the information being processed.

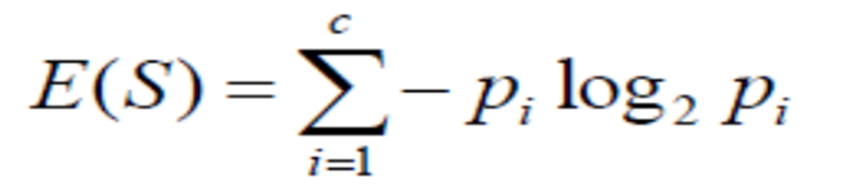
The higher the entropy, the harder it is to draw any conclusions from that information.

The entropy H(X) is zero when the probability is either 0 or 1.

The Entropy is maximum when the probability is 0.5 because it projects perfect randomness in the data and there is no chance if perfectly determining the outcome.

A branch with an entropy of zero is a leaf node and a branch with entropy more than zero needs further splitting.

Mathematically Entropy of attribute is represented as:



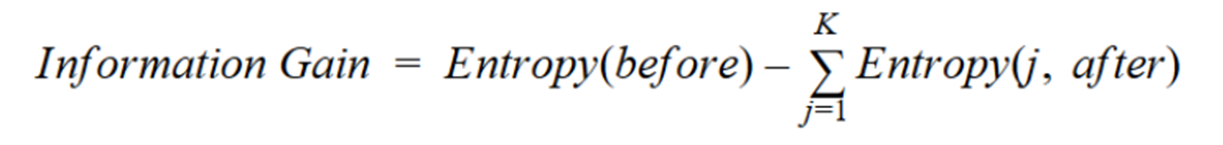
Information Gain

Information gain or IG is a statistical property that measures how well a given attribute separates the training examples according to their target classification.

Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy.

Information gain is a decrease in entropy.

It computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values.



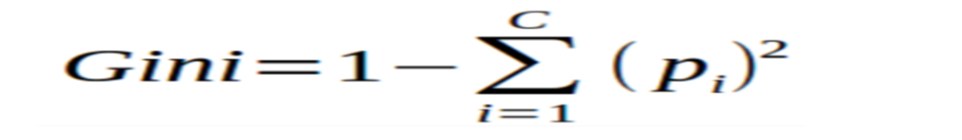
Gini Index  
Gini index, also known as Gini impurity or Gini coefficient, is a measure of impurity or randomness used in decision tree algorithms for classification tasks. It measures the likelihood of a random sample being misclassified based on the distribution of labels in a given set of data.

The Gini index is calculated as follows:

Calculate the probability of each class label in the given set of data.

Square each of the probabilities and sum them.

Subtract the sum from 1.



The resulting value ranges from 0 to 1, with 0 indicating that the data set is perfectly classified and 1 indicating maximum impurity.

In decision tree algorithms, the Gini index is used to determine the best split for a given node, with the split that results in the lowest Gini index being chosen as the optimal one. By recursively splitting the data based on the chosen splits, a decision tree can be built to classify new data points.

Gain ratio

Information gain is biased towards choosing attributes with a large number of values as root nodes.

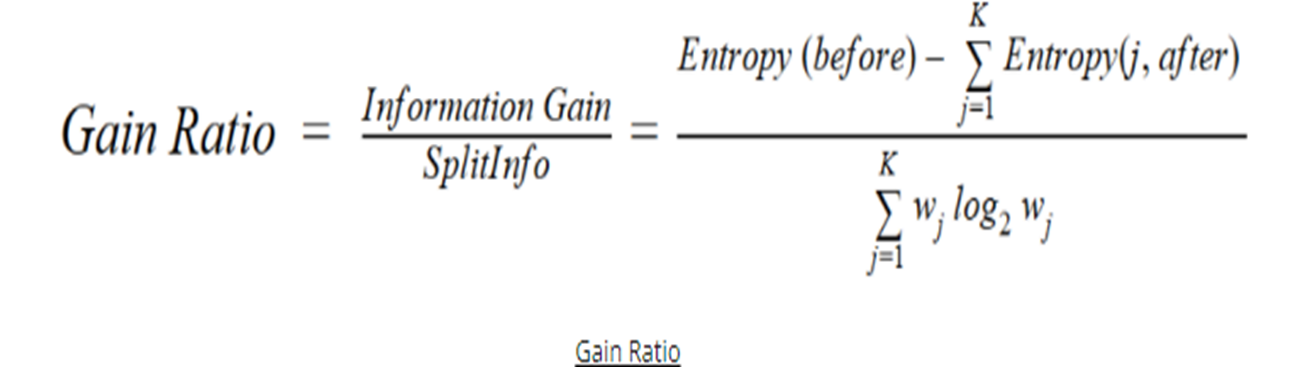
It means it prefers the attribute with a large number of distinct values.

An improvement of ID3, uses Gain ratio which is a modification of Information gain that reduces its bias and is usually the best option.

The attribute with the max gain ratio is used as a splitting attribute.

Gain Ratio or Uncertainty Coefficient is used to normalize the information gain of an attribute against how much entropy that attribute has.

The gain ratio helps to avoid bias towards features with many distinct values, as they may have high information gain simply due to their granularity. Instead, the gain ratio favors features that have a small number of distinct values and can still provide significant information gain.



PRUNING TREES

When decision tree is built many of the branches will reflect anomalies in the training data due to noise or outliers .

A tree that has too many branches and layers can result in overfitting of the training data.

Pruning a decision tree helps to prevent overfitting the training data so that our model generalizes well to unseen data.

Tree pruning methods addresses this problem of over fitting the data.

Uses statistical measures to remove the least reliable branches.

Pruned tree are smaller , faster and better at correctly classifying independent test data as compared to pruned tree.

Pruning a decision tree means to remove a subtree that is redundant and not a useful split and replace it with a leaf node.

Decision tree pruning can be divided into two types: pre-pruning and post-pruning.

Two types of pruning:

Pre-pruning (forward pruning)

Post-pruning (backward pruning)

Pre-pruning

Tree is pruned by halting its construction early.

In pre-pruning, we decide during the building process when to stop adding attributes (based on their information gain)

This is sometimes problematic because Sometimes attributes individually do not contribute much to a decision, but combined, they may have a significant impact

In pre-pruning, we evaluate the pruning condition based on the attribute selection measures at each node.

Examples of pruning conditions include informationGain(Attr)> minGain or treeDepth == MaxDepth. If the condition is satisfied, we prune the subtree.

That means we replace the decision node with a leaf node.

Otherwise, we continue building the tree using our decision tree algorithm.

Pre-pruning has the advantage of being faster and more efficient as it avoids generating overly complex subtrees which overfit the training data.

Post-pruning

Postpruning waits until the full decision tree has built and then prunes the attributes

As the name suggests, post-pruning means to prune after the tree is built.

You grow the tree entirely using your decision tree algorithm and then you prune the subtrees in the tree in a bottom-up fashion.

You start from the bottom decision node and, based on measures such as Gini Impurity or Information Gain, you decide whether to keep this decision node or replace it with a leaf node.

For example, say we want to prune out subtrees that result in least information gain.

When deciding the leaf node, we want to know what leaf our decision tree algorithm would have created if it didn’t create this decision node.

BAYESIAN CLASSIFICATION:

Bayesian classification is a statistical classifier.

They can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

Bayesian classification gives high accuracy and speed when applied to large databases.

Naïve base classifier assumes that the effect of an attribute value on a given class is independent of the values of other attributes. This assumption is called class conditional independence.

The basic idea in NB approaches is To use the joint probabilities of words and categories to estimate the probabilities of categories given a document.

Bayesian classification is a probabilistic classification algorithm that is based on Bayes' theorem, which describes the probability of a hypothesis based on prior knowledge and evidence. It is used to predict the probability of a new data instance belonging to a particular class based on the available training data.

In Bayesian classification, the probability of a new data instance belonging to a particular class is calculated by combining the prior probability of the class with the conditional probability of the instance given the class. The prior probability of a class is the probability of that class occurring in the training data, while the conditional probability of an instance given the class is the probability of the instance having a particular set of feature values given that it belongs to the class.

Bayesian classification can be used with different types of probability distributions, such as the Gaussian distribution, and can handle missing or incomplete data. It is often used in spam filtering, document classification, and medical diagnosis, among other applications.

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.

It is mainly used in text classification that includes a high-dimensional training dataset.

Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

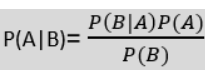
The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:



Where,

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

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Naive Bayes can be used for both binary and multiclass classification problems. It requires a training dataset with labeled examples to learn the conditional probability of each feature given each class. Once the probabilities are calculated, the algorithm can make predictions on new, unseen data by calculating the probability of each class given the features of the data, and selecting the class with the highest probability.

Naive Bayes is commonly used in text classification problems, such as spam filtering and sentiment analysis. It is also used in recommendation systems, medical diagnosis, and image classification.

Advantages of Naïve Bayes Classifier:

Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.

It can be used for Binary as well as Multi-class Classifications.

It performs well in Multi-class predictions as compared to the other Algorithms.

It is the most popular choice for text classification problems.

**Disadvantages** of Naïve Bayes Classifier:

Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

‘

PREDICTION:

Prediction Model makes predictions about future outcomes using historical data combined with statistical modeling,

Predictive analytics uses predictors or known features to create predictive models that will be used in obtaining an output.

Two of the most widely used predictive modeling techniques are regression and neural networks.

Predictive modeling: Predict data values or construct generalized linear models based on the database data.

In regression, accuracy and error measures are used to evaluate the performance of the regression model. Some commonly used accuracy and error measures are:

Mean Absolute Error (MAE): It measures the absolute difference between the predicted and actual values. It is given by the formula:

MAE = (1/n) \* Σ|i=1 to n| |yi - ŷi|

where n is the number of observations, yi is the actual value, and ŷi is the predicted value.

Mean Squared Error (MSE): It measures the squared difference between the predicted and actual values. It is given by the formula:

MSE = (1/n) \* Σ|i=1 to n| (yi - ŷi)^2

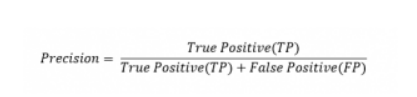
Root Mean Squared Error (RMSE): It is the square root of the mean squared error. It is given by the formula:

RMSE = sqrt(MSE)

R-squared (R2) Coefficient: It measures the proportion of the variation in the dependent variable that is explained by the independent variables. It ranges from 0 to 1, with 1 indicating a perfect fit. It is given by the formula:

PRECISION:

Precision is a measure used in classification tasks that quantifies the percentage of correctly identified positive instances out of all the instances that the classifier labeled as positive. In other words, precision is the number of true positives divided by the sum of true positives and false positives. A high precision indicates a low false positive rate, which means that when the classifier identifies an instance as positive, it is likely to be correct. It is an important metric when the cost of a false positive is high.



An example where precision is important could be a life-saving drug that is being tested for effectiveness. In this scenario, precision would be crucial as it would help ensure that the drug is accurately identified as effective and reliable. If the precision is low, there is a risk of false positives, which would mean that drugs that are not effective may be approved and made available to the public. This could put people's lives at risk as they may rely on the drug to save them when it is not actually effective. Therefore, a high precision is important in ensuring the safety and effectiveness of the drug.

**Recall** is a metric used in machine learning and information retrieval to evaluate how well a system can retrieve relevant information. It measures the proportion of relevant items that have been retrieved over the total number of relevant items in the dataset. In other words, recall is the ability of a model to identify all relevant instances in a dataset.

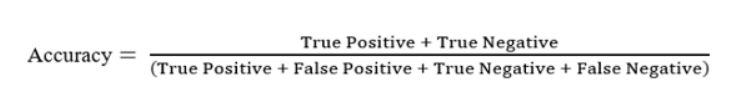
The recall score can be calculated as follows:

Recall = True Positive / (True Positive + False Negative)

where True Positive is the number of correctly identified positive instances, and False Negative is the number of positive instances that were incorrectly identified as negative. The recall score ranges from 0 to 1, with higher values indicating better performance. A perfect recall score of 1 means that the model has identified all relevant instances in the dataset, with no false negatives.

For example, in medical diagnosis, a false negative can be dangerous as it may lead to a patient not receiving necessary treatment. Therefore, in such cases, the recall should be maximized to ensure that all positive instances are correctly identified, even if it means some false positives are classified as positive.

Accuracy is the ratio of the total number of correct predictions and the total number of predictions.



Holdout is a method used in machine learning and statistical modeling to evaluate the performance of a model. It involves dividing a dataset into two subsets: a training set and a testing set. The training set is used to train the model, while the testing set is used to evaluate its performance.

The holdout method is often used in situations where the dataset is large enough to allow for such a division without sacrificing the quality of the training data. The holdout method is also used to prevent overfitting, which occurs when a model is too complex and is trained to fit the training data too closely, resulting in poor performance on new data. By evaluating the model on a separate testing set, it can be determined if the model has learned general patterns in the data or if it is simply memorizing the training data.

Random sampling is a statistical technique that involves selecting a sample of data from a larger population in a random manner. The selection of data is done in such a way that every member of the population has an equal chance of being chosen, and each sample is independent of the other samples. Random sampling is widely used in various fields, including market research, medical research, social sciences, and data analysis, to draw conclusions about the population based on a representative sample. It is a crucial technique in statistics to ensure that the sample data is unbiased and accurately reflects the population's characteristics.

There are four major types of this sampling method, they are;

Simple Random Sampling

In this sampling method, each item in the population has an equal and likely possibility of getting selected in the sample (for example, each member in a group is marked with a specific number). Since the selection of item completely depends on the possibility, therefore this method is called “Method of chance Selection”. Also, the sample size is large, and the item is selected randomly. Thus it is known as “Representative Sampling”.

Systematic Sampling

In this method, the items are chosen from the destination population by choosing the random selecting point and picking the other methods after a fixed sample period. It is equal to the ratio of the total population size and the required population size.

Stratified Sampling

In this sampling method, a population is divided into subgroups to obtain a simple random sample from each group and complete the sampling process (for example, number of girls in a class of 50 strength). These small groups are called strata. The small group is created based on a few features in the population. After dividing the population into smaller groups, the researcher randomly selects the sample.

Clustered Sampling

Cluster sampling is similar to stratified sampling, besides the population is divided into a large number of subgroups (for example, hundreds of thousands of strata or subgroups). After that, some of these subgroups are chosen at random and simple random samples are then gathered within these subgroups. These subgroups are known as clusters. It is basically utilized to lessen the cost of data compilation.

Cross-validation is a technique for validating the model efficiency by training it on the subset of input data and testing on previously unseen subset of the input data. We can also say that it is a technique to check how a statistical model generalizes to an independent dataset.

In machine learning, there is always the need to test the stability of the model. It means based only on the training dataset; we can't fit our model on the training dataset. For this purpose, we reserve a particular sample of the dataset, which was not part of the training dataset. After that, we test our model on that sample before deployment, and this complete process comes under cross-validation. This is something different from the general train-test split.

Hence the basic steps of cross-validations are:

Reserve a subset of the dataset as a validation set.

Provide the training to the model using the training dataset.

Now, evaluate model performance using the validation set. If the model performs well with the validation set, perform the further step, else check for the issues.

**K-fold cross-validation** approach divides the input dataset into K groups of samples of equal sizes. These samples are called folds. For each learning set, the prediction function uses k-1 folds, and the rest of the folds are used for the test set. This approach is a very popular CV approach because it is easy to understand, and the output is less biased than other methods.

The steps for k-fold cross-validation are:

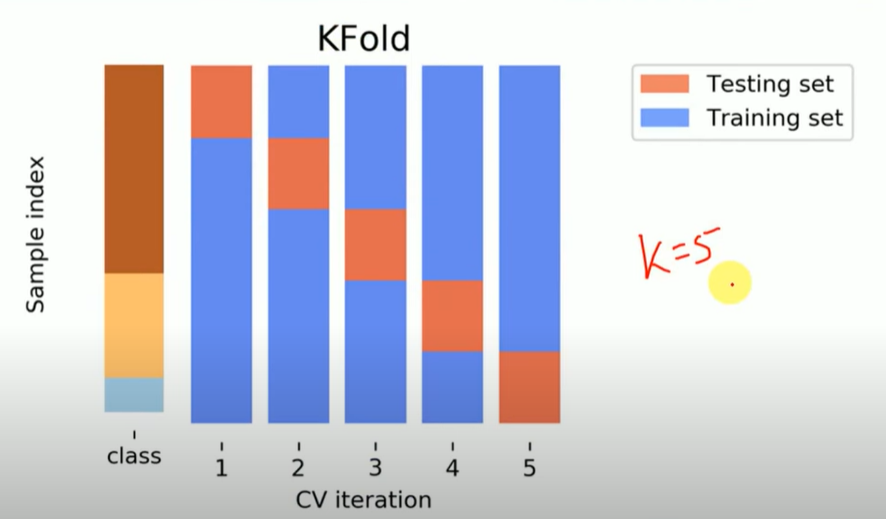
Split the input dataset into K groups

For each group:

Take one group as the reserve or test data set.

Use remaining groups as the training dataset

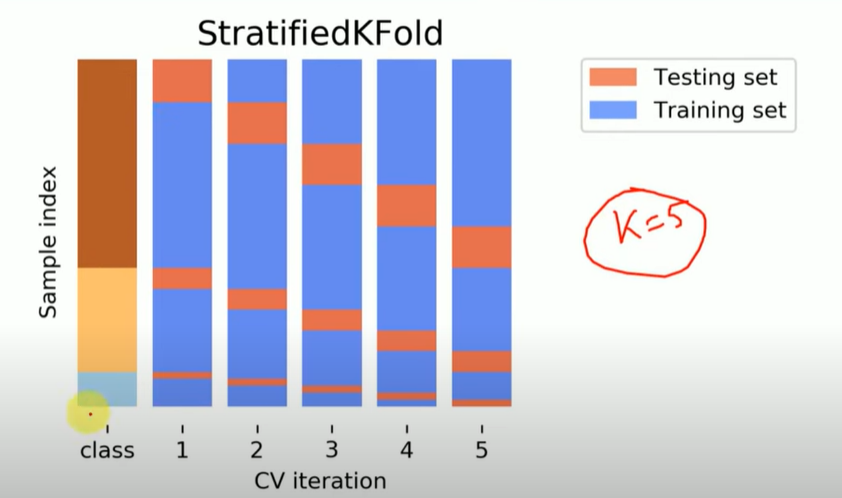
Fit the model on the training set and evaluate the performance of the model using the test set.



Stratified k-fold cross-validation

This technique is similar to k-fold cross-validation with some little changes. This approach works on stratification concept, it is a process of rearranging the data to ensure that each fold or group is a good representative of the complete dataset. To deal with the bias and variance, it is one of the best approaches.

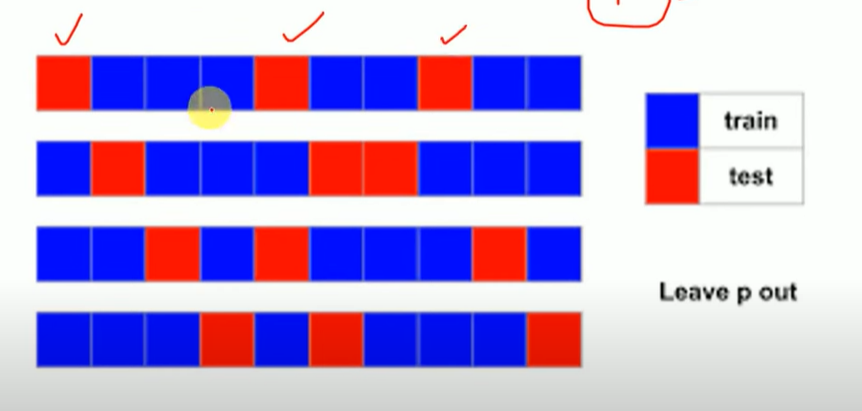
It can be understood with an example of housing prices, such that the price of some houses can be much high than other houses. To tackle such situations, a stratified k-fold cross-validation technique is useful.



Leave-P-out cross-validation

In this approach, the p datasets are left out of the training data. It means, if there are total n datapoints in the original input dataset, then n-p data points will be used as the training dataset and the p data points as the validation set. This complete process is repeated for all the samples, and the average error is calculated to know the effectiveness of the model.

There is a disadvantage of this technique; that is, it can be computationally difficult for the large p.



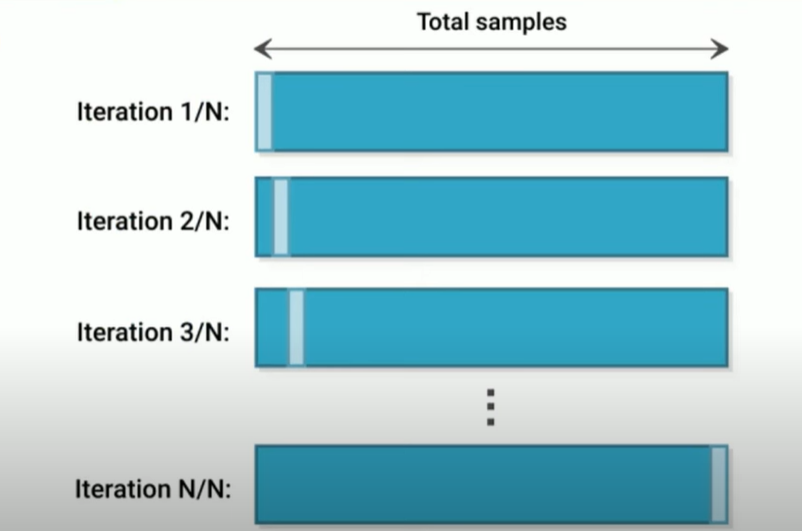
Leave one out cross-validation

This method is similar to the leave-p-out cross-validation, but instead of p, we need to take 1 dataset out of training. It means, in this approach, for each learning set, only one datapoint is reserved, and the remaining dataset is used to train the model. This process repeats for each datapoint. Hence for n samples, we get n different training set and n test set. It has the following features:

In this approach, the bias is minimum as all the data points are used.

The process is executed for n times; hence execution time is high.

This approach leads to high variation in testing the effectiveness of the model as we iteratively check against one data point.

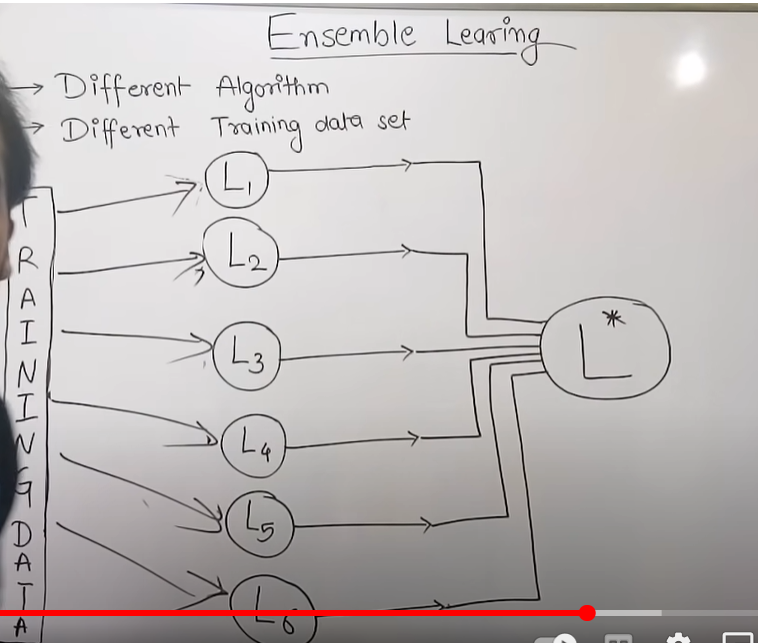


In machine learning, bootstrap refers to the method of generating multiple subsets of the training data by randomly sampling with replacement from the original dataset. The subsets are then used to train multiple models, and the results from these models are combined to make predictions on new data.

The goal of bootstrapping is to improve the accuracy and stability of machine learning models by reducing the risk of overfitting. By training multiple models on different subsets of the data, bootstrapping allows for more robust predictions that are less likely to be biased by a single, outlier datapoint.

One common application of bootstrapping is in the construction of decision trees. By training multiple decision trees on bootstrapped subsets of the data, and combining their predictions using techniques such as bagging or random forests, it is possible to create more accurate and robust models that are less prone to overfitting.

Ensemble learning is a technique in machine learning where multiple models are combined to improve the performance of the overall system. Instead of relying on a single model, ensemble learning uses several models to make more accurate predictions. The idea behind ensemble learning is that by combining the predictions of multiple models, we can reduce the risk of errors and improve the overall accuracy of the system.



There are several ways to perform ensemble learning, including:

Bagging: This technique involves training multiple models on different subsets of the training data and combining their predictions.Bagging (Bootstrap Aggregation) is used to reduce the variance of a decision tree. Suppose a set D of d tuples, at each iteration i, a training set Di of d tuples is sampled with replacement from D (i.e., bootstrap). Then a classifier model Mi is learned for each training set D < i. Each classifier Mi returns its class prediction. The bagged classifier M\* counts the votes and assigns the class with the most votes to X (unknown sample).

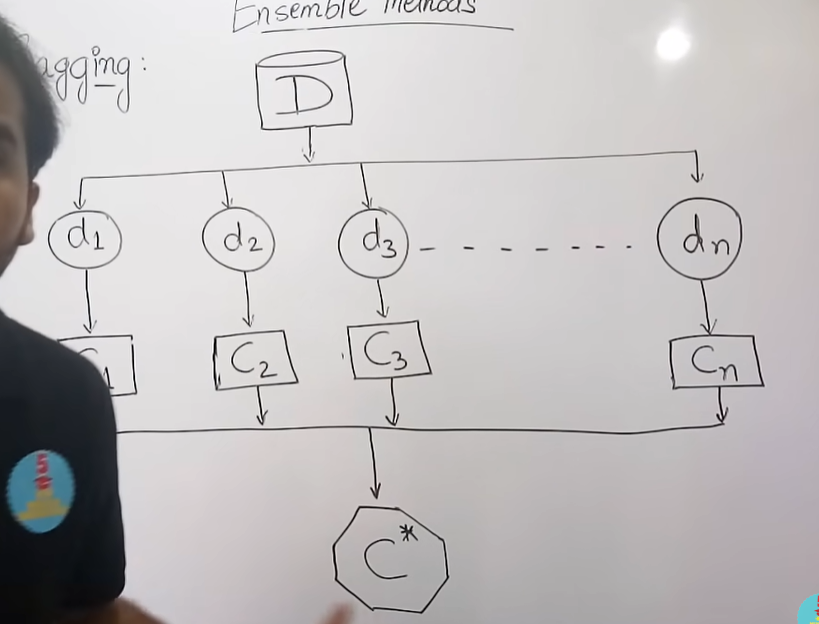
Implementation steps of Bagging –

Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.

A base model is created on each of these subsets.

Each model is learned in parallel from each training set and independent of each other.

The final predictions are determined by combining the predictions from all the models.



Random Forest:

Random Forest is an extension over bagging. Each classifier in the ensemble is a decision tree classifier and is generated using a random selection of attributes at each node to determine the split. During classification, each tree votes and the most popular class is returned.

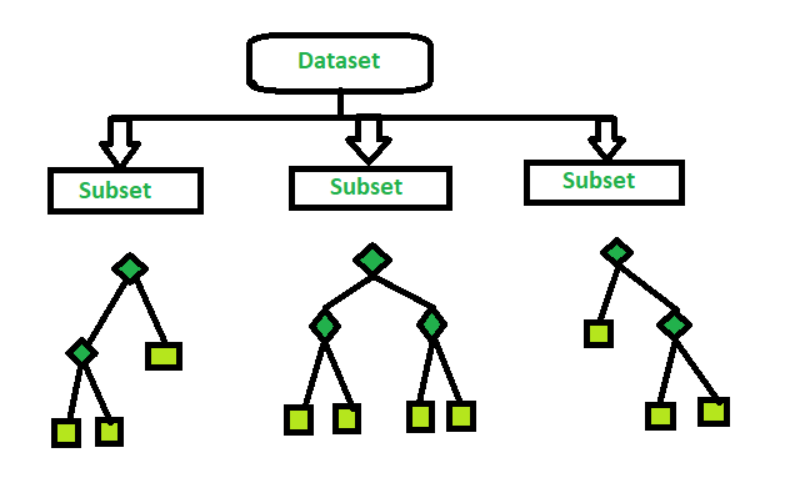
Implementation steps of Random Forest –

Multiple subsets are created from the original data set, selecting observations with replacement.

A subset of features is selected randomly and whichever feature gives the best split is used to split the node iteratively.

The tree is grown to the largest.

Repeat the above steps and prediction is given based on the aggregation of predictions from n number of trees.



Boosting: This technique involves training multiple weak models sequentially, with each new model being trained to improve the errors of the previous models.

Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models is added.

Algorithm:

Initialise the dataset and assign equal weight to each of the data point.

Provide this as input to the model and identify the wrongly classified data points.

Increase the weight of the wrongly classified data points and decrease the weights of correctly classified data points. And then normalize the weights of all data points.

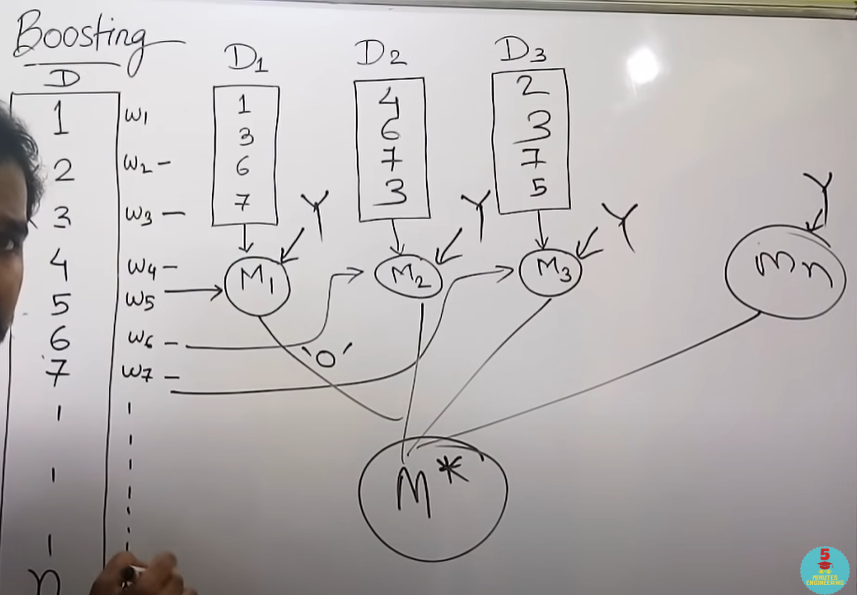
if (got required results)

Goto step 5

else

Goto step 2

End



### Differences Between Bagging and Boosting

| S.NO | Bagging | Boosting |
| --- | --- | --- |
| 1. | The simplest way of combining predictions that  belong to the same type. | A way of combining predictions that  belong to the different types. |
| 2. | Aim to decrease variance, not bias. | Aim to decrease bias, not variance. |
| 3. | Each model receives equal weight. | Models are weighted according to their performance. |
| 4. | Each model is built independently. | New models are influenced  by the performance of previously built models. |
| 5. | Different training data subsets are selected using row sampling with replacement and random sampling methods from the entire training dataset. | Every new subset contains the elements that were misclassified by previous models. |
| 6. | Bagging tries to solve the over-fitting problem. | Boosting tries to reduce bias. |
| 7. | If the classifier is unstable (high variance), then apply bagging. | If the classifier is stable and simple (high bias) the apply boosting. |
| 8. | In this base classifiers are trained parallelly. | In this base classifiers are trained sequentially. |
| 9 | Example: The Random forest model uses Bagging. | Example: The AdaBoost uses Boosting techniques |