1. **In the sense of machine learning, what is a model? What is the best way to train a model?**

**Step 1: Begin with existing data**

Machine learning requires us to have existing data—not the data our application will use when we run it, but data to learn from. You need a lot of real data, in fact, the more the better. The more examples you provide, the better the computer should be able to learn. So just collect every scrap of data you have and dump it and voila! Right?

Wrong. In order to train the computer to understand what we want and what we don’t want, you need to prepare, clean and label your data. Get rid of garbage entries, missing pieces of information, anything that’s ambiguous or confusing. Filter your dataset down to only the information you’re interested in right now. Without high quality data, machine learning does not work. So take your time and pay attention to detail.

**Step 2: Analyze data to identify patterns**

Unlike conventional software development where humans are responsible for interpreting large data sets, with machine learning, you apply a machine learning algorithm to the data. But don’t think you’re off the hook. Choosing the right algorithm, applying it, configuring it and testing it is where the human element comes back in.

There are several platforms to choose from both commercial and open source. Explore solutions from Microsoft, Google, Amazon, IBM or open source frameworks like TensorFlow, Torch and Caffe. They each have their own strengths and downsides, and each will interpret the same dataset a different way. Some are faster to train. Some are more configurable. Some allow for more visibility into the decision process. In order to make the right choice, you need to experiment with a few algorithms and test until you find the one that gives you the results most aligned to what you’re trying to achieve with your data.

When it’s all said and done, and you’ve successfully applied a machine learning algorithm to analyze your data and learn from it, you have a trained model.

**Step 3: Make predictions**

There is so much you can do with your newly trained model. You could import it into a software application you’re building, deploy it into a web back end or upload and host it into a cloud service. Your trained model is now ready to take in new data and feed you predictions, aka results.

These results can look different depending on what kind of algorithm you go with. If you need to know what something is, go with a classification algorithm, which comes in two types. Binary classification categorizes data between two categories. Multi-class classification sorts data between—you guessed it—multiple categories.

When the result you’re looking for is an actual number, you’ll want to use a regression algorithm. Regression takes a lot of different data with different weights of importance and analyzes it with historical data to objectively provide an end result.

Both regression and classification are supervised types of algorithms, meaning you need to provide intentional data and direction for the computer to learn. There is also unsupervised algorithms which don’t require labeled data or any guidance on the kind of result you’re looking for.

One form of unsupervised algorithms is clustering. You use clustering when you want to understand the structure of your data. You provide a set of data and let the algorithm identify the categories within that set. On the other hand, anomaly is an unsupervised algorithm you can use when your data looks normal and uniform, and you want the algorithm to pull anything out of the ordinary that doesn’t fit with the rest of the data.

Although supervised algorithms are more common, it’s good to play around with each algorithm type and use case to better understand probability and practice splitting and training data in different ways. The more you toy with your data, the better your understanding of what machine learning can accomplish will become.

Ultimately, machine learning helps you find new ways to make life easier for your customers and easier for yourself. Self-driving cars not necessary.

1. **In the sense of machine learning, explain the "No Free Lunch" theorem.**

# What is No Free Lunch Theorem

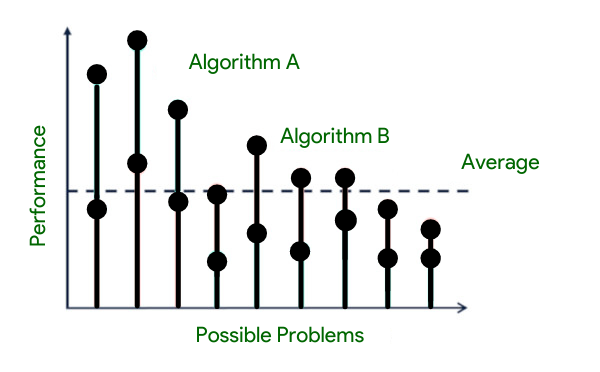
**What is No Free Lunch Theorem :**  
The No Free Lunch Theorem is often used in optimization and machine learning, with little comprehension of what it means or implies.

The theory asserts that when the performance of all optimization methods is averaged across all conceivable problems, they all perform equally well. It indicates that no one optimum optimization algorithm exists. Because of the strong link between optimization, search, and machine learning, there is no one optimum machine learning method for predictive modelling tasks like classification and regression.

They all agree on one point: there is no “best” algorithm for specific kinds of algorithms, since they all perform similarly on average. Mathematically, the computing cost of finding a solution is the same for any solution technique when averaged across all problems in the class. As a result, no solution provides a shortcut.

*There are****two No Free Lunch (NFL) theorems in general:****one for machine learning and one for search and optimization. These two theorems are connected and are frequently combined into a single general postulate (the folklore theorem).*

Although many other scholars have contributed to the collective writings on the No Free Lunch theorems, David Wolpert is the most well-known name connected with these studies.  
Surprisingly, the concept that may have inspired the NFL theorem was first offered by a 1700s philosopher. Yes, you read that correctly! A philosopher, not a mathematician or a statistician..



*Figure 1. Understanding NFL.*

David Hume, a Scottish philosopher, presented the issue of induction in the mid-1700s. This is a philosophical question about whether inductive reasoning leads to true knowledge.

*Inductive reasoning is a type of thinking in which we make inferences about the world based on previous observations.*

According to the “No Free Lunch” theory, there is no one model that works best for every situation. Because the assumptions of a great model for one issue may not hold true for another, it is typical in machine learning to attempt many models to discover the one that performs best for a specific problem. This is especially true in supervised learning, where validation or cross-validation is frequently used to compare the prediction accuracy of many models of various complexity in order to select the optimal model. A good model may also be trained using several methods — for example, linear regression can be learned using normal equations or gradient descent.

**According to the “No Free Lunch” theorem,** all optimization methods perform equally well when averaged over all optimization tasks without re-sampling. This fundamental theoretical notion has had the greatest impact on optimization, search, and supervised learning. The first theorem, No Free Lunch, was rapidly formulated, resulting in a series of research works, which defined a whole field of study with meaningful outcomes across different disciplines of science where the effective exploration of a search region is a vital and crucial activity.

In general, its usefulness is as important as the algorithm. An effective solution is created by matching the utility with the algorithm. If no good conditions for the objective function are known, and one is just working with a black box, no guarantee can be made that this or that method outperforms a (pseudo)random search.

A framework is being created to investigate the relationship between successful optimization algorithms and the issues they solve. A series of “no free lunch” (NFL) theorems are provided, establishing that any improved performance over one class of tasks is compensated by improved performance over another. These theorems provide a geometric explanation of what it means for an algorithm to be well matched to an optimization issue.

*The NFL theorems are also applied to information-theoretic elements of optimization and benchmark measurements of performance.*

There is no such thing as a free lunch, since adding alternatives to a project incurs both direct and opportunity expenses. As a result, incorporating actual alternatives may increase the original development cost. Direct costs are the expenses of additional development effort required to include certain flexibilities into the project’s architecture. Opportunity costs are the expenses of not being able to do anything else (for example, add a feature) as a result of the time and effort spent on generating that flexibility.

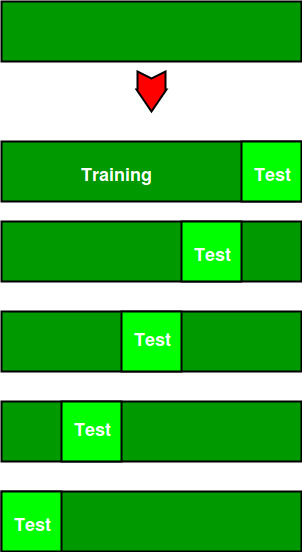
1. **Describe the K-fold cross-validation mechanism in detail.**

### ****K-Fold Cross Validation****

In [K-Fold Cross Validation](https://www.geeksforgeeks.org/k-fold-cross-validation-in-r-programming/), we split the dataset into k number of subsets (known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.

***Note:****It is always suggested that the value of k should be 10 as the lower value of k is takes towards validation and higher value of k leads to LOOCV method.*

#### ****Example**** of K Fold Cross Validation

The diagram below shows an example of the training subsets and evaluation subsets generated in k-fold cross-validation. Here, we have total 25 instances. In first iteration we use the first 20 percent of data for evaluation, and the remaining 80 percent for training ([1-5] testing and [5-25] training) while in the second iteration we use the second subset of 20 percent for evaluation, and the remaining three subsets of the data for training ([5-10] testing and [1-5 and 10-25] training), and so on.

Total instances: 25  
Value of k : 5   
No. Iteration Training set observations Testing set observations  
 1 [ 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24] [0 1 2 3 4]  
 2 [ 0 1 2 3 4 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24] [5 6 7 8 9]  
 3 [ 0 1 2 3 4 5 6 7 8 9 15 16 17 18 19 20 21 22 23 24] [10 11 12 13 14]  
 4 [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 20 21 22 23 24] [15 16 17 18 19]  
 5 [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19] [20 21 22 23 24]

## Comparison between cross-validation and hold out method

### ****Advantages of train/test split:****

1. This runs K times faster than Leave One Out cross-validation because K-fold cross-validation repeats the train/test split K-times.
2. Simpler to examine the detailed results of the testing process.

**Advantages of cross-validation:**

1. More accurate estimate of out-of-sample accuracy.
2. More “efficient” use of data as every observation is used for both training and testing.

## Advantages and Disadvantages of Cross Validation

### Advantages:

1. Overcoming Overfitting: Cross validation helps to prevent overfitting by providing a more robust estimate of the model’s performance on unseen data.
2. Model Selection: Cross validation can be used to compare different models and select the one that performs the best on average.
3. Hyperparameter tuning: Cross validation can be used to optimize the hyperparameters of a model, such as the regularization parameter, by selecting the values that result in the best performance on the validation set.
4. Data Efficient: Cross validation allows the use of all the available data for both training and validation, making it a more data-efficient method compared to traditional validation techniques.

### Disadvantages:

1. Computationally Expensive: Cross validation can be computationally expensive, especially when the number of folds is large or when the model is complex and requires a long time to train.
2. Time-Consuming: Cross validation can be time-consuming, especially when there are many hyperparameters to tune or when multiple models need to be compared.
3. Bias-Variance Tradeoff: The choice of the number of folds in cross validation can impact the bias-variance tradeoff, i.e., too few folds may result in high variance, while too many folds may result in high bias.
4. **Describe the bootstrap sampling method. What is the aim of it?**

# Introduction to Bootstrap plot

Before getting into Bootstrap plot, let us first understand what Bootstrapping (or Bootstrap sampling) is all about.

**Bootstrap Sampling:**It is a method in which we take a sample data repeatedly with replacement from a data set to estimate a population parameter. It is used to determine various parameters of a population.

A bootstrap plot is a graphical representation of the distribution of a statistic calculated from a sample of data. It is often used to visualize the variability and uncertainty of a statistic, such as the mean or standard deviation, by showing the distribution of the statistic over many bootstrapped samples of the data.

In a bootstrap plot, the x-axis represents the values of the statistic and the y-axis represents the frequency of those values. A line is plotted for each bootstrapped sample, with the height of the line indicating the frequency of the statistic’s value in that sample. The distribution of the lines represents the distribution of the statistic over the bootstrapped samples.

The bootstrap plot is a powerful tool for understanding the uncertainty in a statistic, especially when the underlying distribution of the data is unknown or complex. It can also be used to generate confidence intervals for a statistic and to compare the distributions of different statistics.

It is important to note that Bootstrap is a resampling technique which is used to estimate the uncertainty of a statistic from a sample, without making any assumptions about the underlying distribution of the data. It can be used to estimate standard errors, confidence intervals, and to perform hypothesis tests.

**Bootstrap plot:**It is a graphical method used to measure the uncertainty of any desired statistical characteristic of a population. It is an alternative to the confidence interval. (also a mathematical method used for calculation of a statistic).

**Structure**

* **x-axis:** Subsample number.
* **y-axis:** Computed value of the desired statistic for a given subsample.

**Need for a Bootstrap plot:**

Commonly, we can calculate the uncertainty of a statistic of a population mathematically, using confidence intervals. However, in many cases, the uncertainty formula that is derived is mathematically intractable. In such cases, we use the Bootstrap plot.

Suppose, we have 5000 people in a park, and we need to find the average weight of the whole population. It is not feasible to measure the weight of each individual and then take an average of that. This is where bootstrap sampling comes into the picture.

What we do is, we take groups of 5 people randomly from the population and find its mean. We do the same process say 8-10 times. This way, we get a good estimate of the average weight of the population more efficiently.

**Intuition:**

Let us consider an example and understanding how the Bootstrap plot makes it easier to obtain critical information from a large population. Say we have a sample data of 3000 randomly generated uniform numbers. We take out a sub-sample of 30 numbers and find its mean. We do this again for another random sub-sample and so on.

We plot a bootstrap plot of the above-acquired information and just by looking at it, we can easily give a good estimate about the mean of all the 3000 numbers. There is various other useful information one can get out of a bootstrap plot such as:

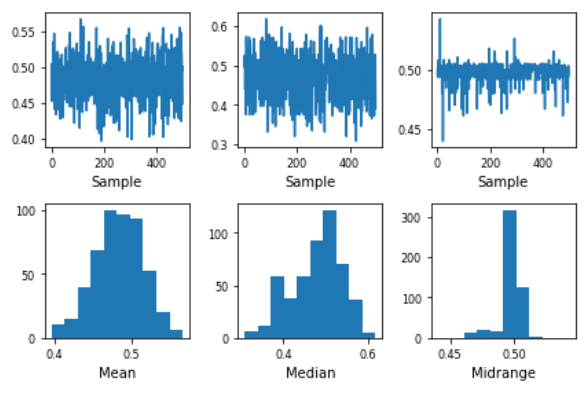
* which sub-sample had the lowest variance, or
* which sub-sample creates the narrowest confidence interval, etc.

**Implementation:**

* Python

|  |
| --- |
| import pandas as pd  import numpy as np    s = pd.Series(np.random.uniform(size=500))  pd.plotting.bootstrap\_plot(s) |

**Output**



**Limitation**

1. The bootstrap plot gives an estimation of the required information from the population, not the exact values.
2. It is highly dependent on the dataset given. It fails to give good results when a lot of subsets have repeated samples.
3. The bootstrap plot becomes ineffective when we are obtaining information that is highly dependent on the tail values. **[As shown in Fig ]**

### Advantages of bootstrap:

* It is a non-parametric method, which means it does not require any assumptions about the underlying distribution of the data.
* It can be used to estimate standard errors and confidence intervals for a wide range of statistics.
* It can be used to estimate the uncertainty of a statistic even when the sample size is small.
* It can be used to perform hypothesis tests and compare the distributions of different statistics.
* It is widely used in many fields such as statistics, finance, and machine learning

### Disadvantages of bootstrap:

* It can be computationally intensive, especially when working with large datasets.
* It may not be appropriate for all types of data, such as highly skewed or heavy-tailed distributions.
* It may not be appropriate for estimating the uncertainty of statistics that have very large variances.
* It may not be appropriate for estimating the uncertainty of statistics that are not smooth or have very different variances.
* It may not always be a good substitute for other statistical methods like asymptotic methods, when large sample sizes are available.

1. **What is the significance of calculating the Kappa value for a classification model? Demonstrate how to measure the Kappa value of a classification model using a sample collection of results.**

**Kappa score is a famous evaluation metric for measuring the performance of classification problems in machine learning. It was proposed by a famous statistician Jacob Cohen and later it was accepted globally.**

Let’s understand more about these metrics.

Kappa score is based on a principle called **“Agreement”**.

**The agreement is nothing but coming to the same opinion by 2 different human beings**.

For example, consider a conversation with 2 persons-

**James**: “Titanic is an excellent movie. I like to watch it every time when it is getting telecasted on TV”.

**Alice**:” Yes. Titanic is a gem. Especially the story and screenplay are outstanding”.

Here, Both James and Alice have the same opinion regarding the movie that means their agreement is strong.

Now, consider another scenario-

**James**: “I think Indian movies are much better than American movies because they usually blend with our sentiments and make us repeated watchers. The story's sense of Indian movies is very nice”.

**Alice**: “No. Indian movies can never match up with the grandeur and viewing experience of Hollywood movies”.

Here, they both have different thought processes. James is a fan of Indian movies while Alice is fond of Hollywood movies. Hence, their agreement is weak.

Now let us understand about the Chance of agreement-

**“John is a student who is dedicatedly studying for 5 hours every day. In all the examinations, he is getting good marks.”**

Hence, it is explicitly visible that there is a strong relationship between the number of hours spending for studying a subject and marks getting in the examinations. So, anyone can dedicatedly spend more hours studying the subject and get good marks.

**“John walks above a bridge every day for getting into the school. Most of the time, he ends up dropping his books down when he enters the bridge.”**

Does this mean there is a correlation between bridge and book? No. Actually, it might have happened because of some random chances.

* He may be a hypersensitive person who is fear of walking above the bridge so that he may be dropping the book every time.
* His bag which is containing his book maybe not properly aligned which makes it let down coincidentally when he is entering the bridge.
* The book may be a very light-weighted one that immediately falls down when some wind is coming above the bridge.

Hence, all of these things are not mathematically explainable. They are just random chances.

**Similarly, the Chance of agreement is the probability of 2 people will having an opinion on a subject purely based on random chances but not because of any mathematical fact.**

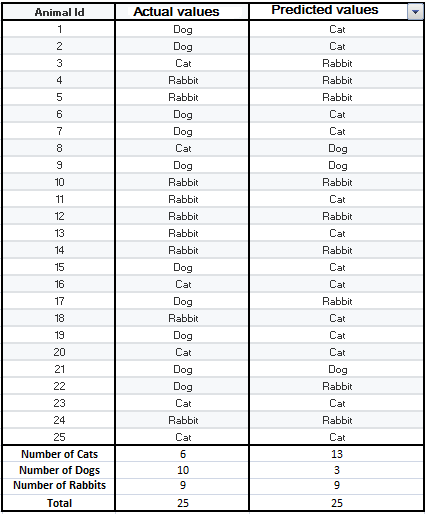
I hope now you understood about Agreement a Chance of Agreement. Let’s now mathematically understand how the Kappa score is used in multi-class classification problems.

Consider that we have a machine learning problem where we need to identify the name of the animal. There are 3 possible classes-

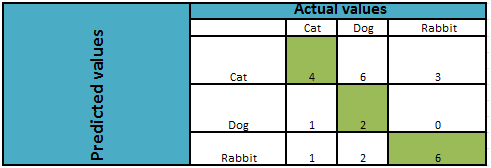
* Dog
* Cat
* Rabbit



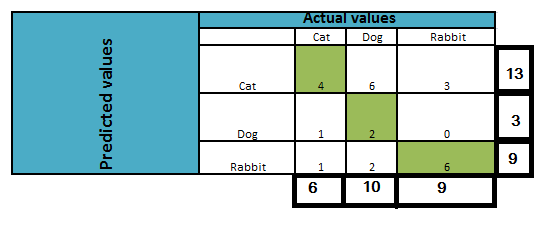
After training the model, let’s assume that we got the following output as predicted values-



Now, we need to summarize the table as a matrix like this-



* Each row shows the predictions done by the model.
* Each column shows the actual values in the data.



**The value of the Agreement will be the sum of the diagonal cells of the matrix (the cells mentioned in green color) divided by the total number of observations.**

Agreement = 4 + 2 + 6 = 12/25

Agreement = 0.48

Now let’s calculate the value of the Chance of Agreement

Mathematically,

**Chance of Agreement = ProbabilityA (Agreement) X ProbabilityB (Agreement)**

Here,

Prob A ( Cat ) = 6/25=0.24 ( The number of observations whose actual value is Cat in the input data set)

Prob B( Cat )=13/25=0.5224 ( The number of observations predicted as Cat by the model)

**Chance of Agreement (Cat) = 0.24 X 0.5524 = 0.12**

Prob A ( Dog ) = 10/25=0.4 (The number of observations whose actual value is Dog in the input data set)

Prob B( Dog )=3/25=0.12 ( The number of observations predicted as Dog by the model)

**Chance of Agreement (Dog) = 0.4 X 0.1248 = 0.048**

Prob A ( Rabbit ) = 9/25=0.36 (The number of observations whose actual value is Rabbit in the input data set)

Prob B( Rabbit )=9 /25=0.36 ( The number of observations predicted as Rabbit by the model)

**Chance of Agreement (Rabbit) = 0.36 X 0.36 = 0.118**

**Chance of Agreement = Chance of Agreement(Cat) + Chance of Agreement(Dog) + Chance of Agreement(Rabbit)**

**Chance of Agreement = 0.1248 +0.0480+0.1188=0.3024**

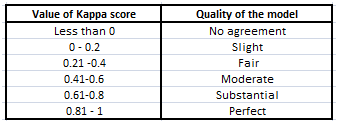
Now, we can find the Kappa score using the following formula-

**Kappa Score= (Agreement-Chance of Agreement)/ (1- Chance of Agreement)**

Here,

Kappa Score = (0.48–0.3024)/(1–0.3024)  
**Kappa Score = 0.2546**

**The scale of Kappa value interpretation according to the research paper “The Measurement of Observer Agreement for Categorical Data” by J. Richard Landis and Gary G. Koch (1977) is given in the following table-**



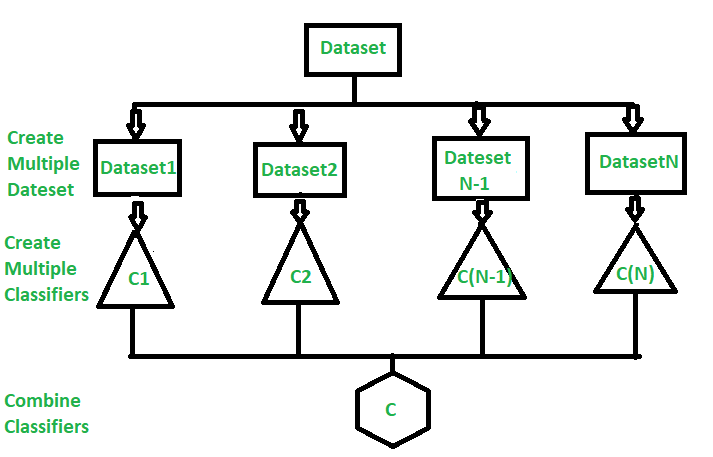
It is one of the basic goodness of fitmetrics that in most of the time range around 0 to 1. The higher the value of the Kappa score examined as higher the coherence and predictive ability of the model.

1. **Describe the model ensemble method. In machine learning, what part does it play?**

# Ensemble Classifier | Data Mining

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote.

***Advantage :****Improvement in predictive accuracy.****Disadvantage :****It is difficult to understand an ensemble of classifiers.*



**Why do ensembles work?**

Dietterich(2002) showed that ensembles overcome three problems –

* **Statistical Problem –**  
  The Statistical Problem arises when the hypothesis space is too large for the amount of available data. Hence, there are many hypotheses with the same accuracy on the data and the learning algorithm chooses only one of them! There is a risk that the accuracy of the chosen hypothesis is low on unseen data!
* **Computational Problem –**  
  The Computational Problem arises when the learning algorithm cannot guarantees finding the best hypothesis.
* **Representational Problem –**  
  The Representational Problem arises when the hypothesis space does not contain any good approximation of the target class(es).

**Main Challenge for Developing Ensemble Models?**

The main challenge is not to obtain highly accurate base models, but rather to obtain base models which make different kinds of errors. For example, if ensembles are used for classification, high accuracies can be accomplished if different base models misclassify different training examples, even if the base classifier accuracy is low.

***Methods for Independently Constructing Ensembles –***

* *Majority Vote*
* *Bagging and Random Forest*
* *Randomness Injection*
* *Feature-Selection Ensembles*
* *Error-Correcting Output Coding*

***Methods for Coordinated Construction of Ensembles –***

* *Boosting*
* *Stacking*

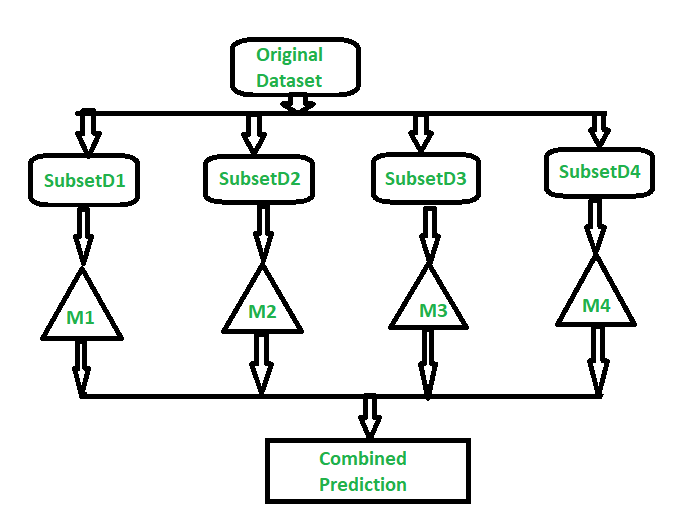
***Reliable Classification:****Meta-Classifier Approach****Co-Training and Self-Training***

**Types of Ensemble Classifier –**

**Bagging:**  
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 –**

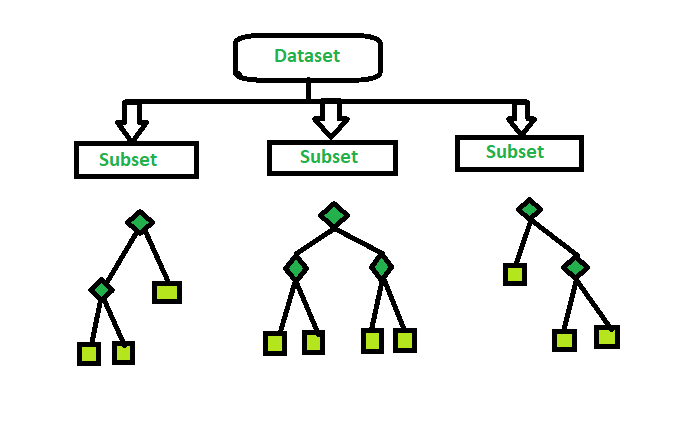
1. Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
2. A base model is created on each of these subsets.
3. Each model is learned in parallel from each training set and independent of each other.
4. The final predictions are determined 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 –**

* 1. Multiple subsets are created from the original data set, selecting observations with replacement.
  2. A subset of features is selected randomly and whichever feature gives the best split is used to split the node iteratively.
  3. The tree is grown to the largest.
  4. Repeat the above steps and prediction is given based on the aggregation of predictions from n number of trees.



1. **What is a descriptive model's main purpose? Give examples of real-world problems that descriptive models were used to solve.**

A **descriptive model** describes a system or other entity and its relationship to its environment. It is generally used to help specify and/or understand what the system is, what it does, and how it does it.

A **geometric model** or **spatial model** is a descriptive model that represents geometric and/or spatial relationships. Mechanical three-dimensional computer aided design (CAD) models are geometric models that include detailed information, including dimensions, tolerances, and other descriptive data such as material characteristics. A 3D representation of land topography and other features that are often presented as maps and other visualizations is also a kind of spatial model.

A **logical model** is a descriptive model that primarily represents logical relationships and dependencies such as functional, connectivity, and traceability relationships. Examples of logical models include a circuit design model that describes electrical components and their interconnections, and a model of system composition such as a bill-of-materials.

The **system model** is a logical model that is introduced in Chapter 2 Section 2.1.2. This model captures the requirements, structure, behavior, and parametric constraints associated with a system and its environment, along with the relationships between these elements. As discussed throughout this book, SysML is a modeling language used to capture the system model. SysML supports various abstraction techniques and provides the ability to represent many different views of the system, such as a black-box view, white-box view, and a security view. The system model can also be queried and analyzed for different purposes, such as providing traceability analysis, assessing the completeness of the model, and validating model correctness.

1. **Describe how to evaluate a linear regression model.**

# Linear Regression in Machine learning

**Machine Learning** is a branch of Artificial intelligence that focuses on the development of algorithms and statistical models that can learn from and make predictions on data. **Linear regression** is also a type of machine-learning algorithm more specifically a **supervised machine-learning algorithm** that learns from the labelled datasets and maps the data points to the most optimized linear functions. which can be used for prediction on new datasets.

First of we should know what supervised machine learning algorithms is. It is a type of machine learning where the algorithm learns from labelled data.  Labeled data means the dataset whose respective target value is already known. Supervised learning has two types:

* **Classification**: It predicts the class of the dataset based on the independent input variable. Class is the categorical or discrete values. like the image of an animal is a cat or dog?
* **Regression**: It predicts the continuous output variables based on the independent input variable. like the prediction of house prices based on different parameters like house age, distance from the main road, location, area, etc.

Here, we will discuss one of the simplest types of regression i.e. **Linear Regression.**

**Table of Content**

* [What is Linear Regression?](https://www.geeksforgeeks.org/ml-linear-regression/#what-is-linear-regression)
* [Types of Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/#types-of-linear-regression)
* [What is the best Fit Line?](https://www.geeksforgeeks.org/ml-linear-regression/#what-is-the-best-fit-line)
* [Cost function for Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/#cost-function-for-linear-regression)
* [Assumptions of Simple Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/#assumptions-of-simple-linear-regression)
* [Assumptions of Multiple Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/#assumptions-of-multiple-linear-regression)
* [Evaluation Metrics for Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/#evaluation-metrics-for-linear-regression)
* [Python Implementation of Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/#python-implementation-of-linear-regression)
* [Regularization Techniques for Linear Models](https://www.geeksforgeeks.org/ml-linear-regression/#regularization-techniques-for-linear-models)
* [Applications of Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/#applications-of-linear-regression)
* [Advantages & Disadvantages of Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/#advantages-disadvantages-of-linear-regression)
* [Linear Regression – Frequently Asked Questions (FAQs)](https://www.geeksforgeeks.org/ml-linear-regression/#linear-regression-frequently-asked-questions-faqs)

## ****What is Linear Regression?****

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

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

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

### Why Linear Regression is Important?

The interpretability of linear regression is a notable strength. The model’s equation provides clear coefficients that elucidate the impact of each independent variable on the dependent variable, facilitating a deeper understanding of the underlying dynamics. Its simplicity is a virtue, as linear regression is transparent, easy to implement, and serves as a foundational concept for more complex algorithms.

Linear regression is not merely a predictive tool; it forms the basis for various advanced models. Techniques like regularization and support vector machines draw inspiration from linear regression, expanding its utility. Additionally, linear regression is a cornerstone in assumption testing, enabling researchers to validate key assumptions about the data.

## Types of Linear Regression

There are two main types of linear regression:

### ****Simple Linear Regression****

This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear regression is:  
𝑦=𝛽0+𝛽1𝑋*y*=*β*0​+*β*1​*X*  
where:

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

### ****Multiple Linear Regression****

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

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

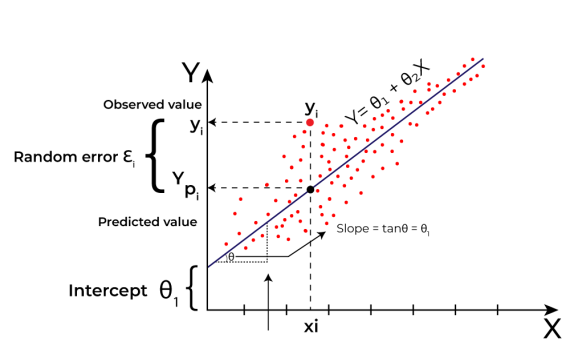
#### The goal of the algorithm is to find the ****best Fit Line**** equation that can predict the values based on the independent variables.

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

## What is the best Fit Line?

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

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



*Linear Regression*

Here Y is called a dependent or target variable and X is called an independent variable also known as the predictor of Y. There are many types of functions or modules that can be used for regression. A linear function is the simplest type of function. Here, X may be a single feature or multiple features representing the problem.

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

We utilize the cost function to compute the best values in order to get the best fit line since different values for weights or the coefficient of lines result in different regression lines.

### ****Hypothesis function in Linear Regression****

As we have assumed earlier that our independent feature is the experience i.e X and the respective salary Y is the dependent variable. Let’s assume there is a linear relationship between X and Y then the salary can be predicted using:

𝑌^=𝜃1+𝜃2𝑋*Y*^=*θ*1​+*θ*2​*X*

OR

𝑦^𝑖=𝜃1+𝜃2𝑥𝑖*y*^​*i*​=*θ*1​+*θ*2​*xi*​

Here,

* 𝑦𝑖𝜖𝑌(𝑖=1,2,⋯,𝑛)     *yi*​*ϵY*(*i*=1,2,⋯,*n*)  are labels to data (Supervised learning)
* 𝑥𝑖𝜖𝑋(𝑖=1,2,⋯,𝑛)     *xi*​*ϵX*(*i*=1,2,⋯,*n*)  are the input independent training data (univariate – one input variable(parameter))
* 𝑦𝑖^𝜖𝑌^(𝑖=1,2,⋯,𝑛)     *yi*​^​*ϵY*^(*i*=1,2,⋯,*n*)  are the predicted values.

The model gets the best regression fit line by finding the best θ1 and θ2 values.

* **θ1:** intercept
* **θ2:** coefficient of x

Once we find the best θ1 and θ2 values, we get the best-fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

### ****How to update θ1 and θ2 values to get the best-fit line?****

To achieve the best-fit regression line, the model aims to predict the target value 𝑌^     *Y*^  such that the error difference between the predicted value 𝑌^     *Y*^  and the true value Y is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimizes the error between the predicted y value (pred) and the true y value (y).

𝑚𝑖𝑛𝑖𝑚𝑖𝑧𝑒1𝑛∑𝑖=1𝑛(𝑦𝑖^−𝑦𝑖)2*minimizen*1​∑*i*=1*n*​(*yi*​^​−*yi*​)2

## Cost function for Linear Regression

The [cost function](https://www.geeksforgeeks.org/what-is-cost-function/) or the[loss function](https://www.geeksforgeeks.org/ml-common-loss-functions/) is nothing but the error or difference between the predicted value 𝑌^     *Y*^  and the true value Y.

In Linear Regression, the **Mean Squared Error (MSE)** cost function is employed, which calculates the average of the squared errors between the predicted values 𝑦^𝑖*y*^​*i*​ and the actual values 𝑦𝑖*yi*​. The purpose is to determine the optimal values for the intercept 𝜃1*θ*1​ and the coefficient of the input feature 𝜃2*θ*2​ providing the best-fit line for the given data points. The linear equation expressing this relationship is 𝑦^𝑖=𝜃1+𝜃2𝑥𝑖*y*^​*i*​=*θ*1​+*θ*2​*xi*​.

MSE function can be calculated as:

Cost function(𝐽)=1𝑛∑𝑛𝑖(𝑦𝑖^−𝑦𝑖)2Cost function(*J*)=*n*1​∑*ni*​(*yi*​^​−*yi*​)2

Utilizing the MSE function, the iterative process of gradient descent is applied to update the values of \𝜃1&𝜃2*θ*1​&*θ*2​. This ensures that the MSE value converges to the global minima, signifying the most accurate fit of the linear regression line to the dataset.

This process involves continuously adjusting the parameters \(\theta\_1\) and \(\theta\_2\) based on the gradients calculated from the MSE. The final result is a linear regression line that minimizes the overall squared differences between the predicted and actual values, providing an optimal representation of the underlying relationship in the data.

### ****Gradient Descent for Linear Regression****

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

A gradient is nothing but a derivative that defines the effects on outputs of the function with a little bit of variation in inputs.

Let’s differentiate the cost function(J) with respect to 𝜃1     *θ*1​

𝐽’𝜃1=∂𝐽(𝜃1,𝜃2)∂𝜃1=∂∂𝜃1[1𝑛(∑𝑖=1𝑛(𝑦^𝑖−𝑦𝑖)2)]=1𝑛[∑𝑖=1𝑛2(𝑦^𝑖−𝑦𝑖)(∂∂𝜃1(𝑦^𝑖−𝑦𝑖))]=1𝑛[∑𝑖=1𝑛2(𝑦^𝑖−𝑦𝑖)(∂∂𝜃1(𝜃1+𝜃2𝑥𝑖−𝑦𝑖))]=1𝑛[∑𝑖=1𝑛2(𝑦^𝑖−𝑦𝑖)(1+0−0)]=1𝑛[∑𝑖=1𝑛(𝑦^𝑖−𝑦𝑖)(2)]=2𝑛∑𝑖=1𝑛(𝑦^𝑖−𝑦𝑖)*J*’*θ*1​​​=∂*θ*1​∂*J*(*θ*1​,*θ*2​)​=∂*θ*1​∂​[*n*1​(*i*=1∑*n*​(*y*^​*i*​−*yi*​)2)]=*n*1​[*i*=1∑*n*​2(*y*^​*i*​−*yi*​)(∂*θ*1​∂​(*y*^​*i*​−*yi*​))]=*n*1​[*i*=1∑*n*​2(*y*^​*i*​−*yi*​)(∂*θ*1​∂​(*θ*1​+*θ*2​*xi*​−*yi*​))]=*n*1​[*i*=1∑*n*​2(*y*^​*i*​−*yi*​)(1+0−0)]=*n*1​[*i*=1∑*n*​(*y*^​*i*​−*yi*​)(2)]=*n*2​*i*=1∑*n*​(*y*^​*i*​−*yi*​)​

Let’s differentiate the cost function(J) with respect to 𝜃2*θ*2​

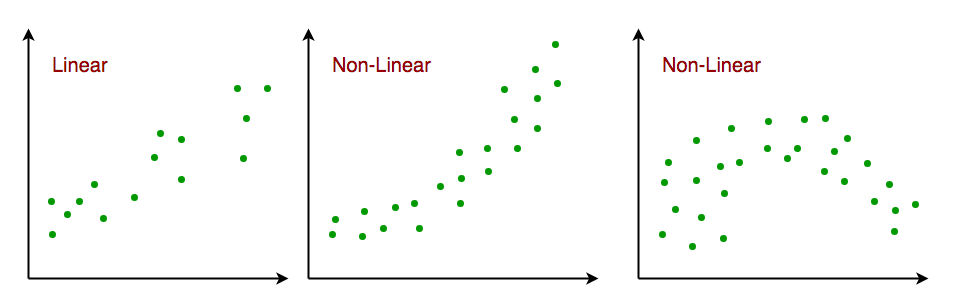
𝐽’𝜃2=∂𝐽(𝜃1,𝜃2)∂𝜃2=∂∂𝜃2[1𝑛(∑𝑖=1𝑛(𝑦^𝑖−𝑦𝑖)2)]=1𝑛[∑𝑖=1𝑛2(𝑦^𝑖−𝑦𝑖)(∂∂𝜃2(𝑦^𝑖−𝑦𝑖))]=1𝑛[∑𝑖=1𝑛2(𝑦^𝑖−𝑦𝑖)(∂∂𝜃2(𝜃1+𝜃2𝑥𝑖−𝑦𝑖))]=1𝑛[∑𝑖=1𝑛2(𝑦^𝑖−𝑦𝑖)(0+𝑥𝑖−0)]=1𝑛[∑𝑖=1𝑛(𝑦^𝑖−𝑦𝑖)(2𝑥𝑖)]=2𝑛∑𝑖=1𝑛(𝑦^𝑖−𝑦𝑖)⋅𝑥𝑖*J*’*θ*2​​​=∂*θ*2​∂*J*(*θ*1​,*θ*2​)​=∂*θ*2​∂​[*n*1​(*i*=1∑*n*​(*y*^​*i*​−*yi*​)2)]=*n*1​[*i*=1∑*n*​2(*y*^​*i*​−*yi*​)(∂*θ*2​∂​(*y*^​*i*​−*yi*​))]=*n*1​[*i*=1∑*n*​2(*y*^​*i*​−*yi*​)(∂*θ*2​∂​(*θ*1​+*θ*2​*xi*​−*yi*​))]=*n*1​[*i*=1∑*n*​2(*y*^​*i*​−*yi*​)(0+*xi*​−0)]=*n*1​[*i*=1∑*n*​(*y*^​*i*​−*yi*​)(2*xi*​)]=*n*2​*i*=1∑*n*​(*y*^​*i*​−*yi*​)⋅*xi*​​

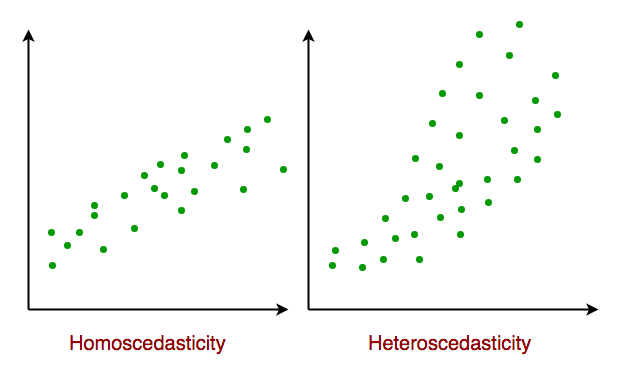
Finding the coefficients of a linear equation that best fits the training data is the objective of linear regression. By moving in the direction of the Mean Squared Error negative gradient with respect to the coefficients, the coefficients can be changed. And the respective intercept and coefficient of X will be if 𝛼     *α*  is the learning rate.

𝜃1=𝜃1–𝛼(𝐽’𝜃1)=𝜃1−𝛼(2𝑛∑𝑖=1𝑛(𝑦^𝑖−𝑦𝑖))𝜃2=𝜃2–𝛼(𝐽’𝜃2)=𝜃2–𝛼(2𝑛∑𝑖=1𝑛(𝑦^𝑖−𝑦𝑖)⋅𝑥𝑖)*θ*1​​=*θ*1​–*α*(*J*’*θ*1​​)=*θ*1​−*α*(*n*2​*i*=1∑*n*​(*y*^​*i*​−*yi*​))​*θ*2​​=*θ*2​–*α*(*J*’*θ*2​​)=*θ*2​–*α*(*n*2​*i*=1∑*n*​(*y*^​*i*​−*yi*​)⋅*xi*​)​

## Assumptions of Simple Linear Regression

Linear regression is a powerful tool for understanding and predicting the behavior of a variable, however, it needs to meet a few conditions in order to be accurate and dependable solutions.

1. **Linearity**: The independent and dependent variables have a linear relationship with one another. This implies that changes in the dependent variable follow those in the independent variable(s) in a linear fashion. This means that there should be a straight line that can be drawn through the data points. If the relationship is not linear, then linear regression will not be an accurate model.  
   
2. **Independence**: The observations in the dataset are independent of each other. This means that the value of the dependent variable for one observation does not depend on the value of the dependent variable for another observation. If the observations are not independent, then linear regression will not be an accurate model.
3. **Homoscedasticity**: Across all levels of the independent variable(s), the variance of the errors is constant. This indicates that the amount of the independent variable(s) has no impact on the variance of the errors. If the variance of the residuals is not constant, then linear regression will not be an accurate model.



*Homoscedasticity in Linear Regression*

1. **Normality**: The residuals should be normally distributed. This means that the residuals should follow a bell-shaped curve. If the residuals are not normally distributed, then linear regression will not be an accurate model.

## Assumptions of Multiple Linear Regression

For Multiple Linear Regression, all four of the assumptions from Simple Linear Regression apply. In addition to this, below are few more:

1. **No multicollinearity**: There is no high correlation between the independent variables. This indicates that there is little or no correlation between the independent variables. Multicollinearity occurs when two or more independent variables are highly correlated with each other, which can make it difficult to determine the individual effect of each variable on the dependent variable. If there is multicollinearity, then multiple linear regression will not be an accurate model.
2. **Additivity:** The model assumes that the effect of changes in a predictor variable on the response variable is consistent regardless of the values of the other variables. This assumption implies that there is no interaction between variables in their effects on the dependent variable.
3. **Feature Selection:** In multiple linear regression, it is essential to carefully select the independent variables that will be included in the model. Including irrelevant or redundant variables may lead to overfitting and complicate the interpretation of the model.
4. **Overfitting:** Overfitting occurs when the model fits the training data too closely, capturing noise or random fluctuations that do not represent the true underlying relationship between variables. This can lead to poor generalization performance on new, unseen data.

### ****Multicollinearity****

[Multicollinearity](https://www.geeksforgeeks.org/multicollinearity-in-data/) is a statistical phenomenon that occurs when two or more independent variables in a multiple regression model are highly correlated, making it difficult to assess the individual effects of each variable on the dependent variable.

**Detecting Multicollinearity includes two techniques:**

* **Correlation Matrix:** Examining the correlation matrix among the independent variables is a common way to detect multicollinearity. High correlations (close to 1 or -1) indicate potential multicollinearity.
* **VIF (Variance Inflation Factor):** VIF is a measure that quantifies how much the variance of an estimated regression coefficient increases if your predictors are correlated. A high VIF (typically above 10) suggests multicollinearity.

## Evaluation Metrics for Linear Regression

A variety of [evaluation measures](https://www.geeksforgeeks.org/metrics-for-machine-learning-model/) can be used to determine the strength of any linear regression model. These assessment metrics often give an indication of how well the model is producing the observed outputs.

The most common measurements are:

### Mean Square Error (MSE)

[Mean Squared Error (MSE)](https://www.geeksforgeeks.org/python-mean-squared-error/) is an evaluation metric that calculates the average of the squared differences between the actual and predicted values for all the data points. The difference is squared to ensure that negative and positive differences don’t cancel each other out.

𝑀𝑆𝐸=1𝑛∑𝑖=1𝑛(𝑦𝑖–𝑦𝑖^)2*MSE*=*n*1​∑*i*=1*n*​(*yi*​–*yi*​​)2

Here,

* n is the number of data points.
* yi is the actual or observed value for the ith data point.
* 𝑦𝑖^*yi*​​ is the predicted value for the ith data point.

MSE is a way to quantify the accuracy of a model’s predictions. MSE is sensitive to outliers as large errors contribute significantly to the overall score.

### Mean Absolute Error (MAE)

[Mean Absolute Error](https://www.geeksforgeeks.org/how-to-calculate-mean-absolute-error-in-python/)is an evaluation metric used to calculate the accuracy of a regression model. MAE measures the average absolute difference between the predicted values and actual values.

Mathematically, MAE is expressed as:

𝑀𝐴𝐸=1𝑛∑𝑖=1𝑛∣𝑌𝑖–𝑌𝑖^∣*MAE*=*n*1​∑*i*=1*n*​∣*Yi*​–*Yi*​​∣

Here,

* n is the number of observations
* Yi represents the actual values.
* 𝑌𝑖^*Yi*​​ represents the predicted values

Lower MAE value indicates better model performance. It is not sensitive to the outliers as we consider absolute differences.

### ****Root Mean Squared Error (RMSE)****

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

In mathematical notation, it can be expressed as:  
𝑅𝑀𝑆𝐸=𝑅𝑆𝑆𝑛=∑𝑖=2𝑛(𝑦𝑖𝑎𝑐𝑡𝑢𝑎𝑙−𝑦𝑖𝑝𝑟𝑒𝑑𝑖𝑐𝑡𝑒𝑑)2𝑛*RMSE*=*nRSS*​​=*n*∑*i*=2*n*​(*yiactual*​−*yipredicted*​)2​​  
Rather than dividing the entire number of data points in the model by the number of degrees of freedom, one must divide the sum of the squared residuals to obtain an unbiased estimate. Then, this figure is referred to as the Residual Standard Error (RSE).

In mathematical notation, it can be expressed as:  
𝑅𝑀𝑆𝐸=𝑅𝑆𝑆𝑛=∑𝑖=2𝑛(𝑦𝑖𝑎𝑐𝑡𝑢𝑎𝑙−𝑦𝑖𝑝𝑟𝑒𝑑𝑖𝑐𝑡𝑒𝑑)2(𝑛−2)*RMSE*=*nRSS*​​=(*n*−2)∑*i*=2*n*​(*yiactual*​−*yipredicted*​)2​​

RSME is not as good of a metric as R-squared. Root Mean Squared Error can fluctuate when the units of the variables vary since its value is dependent on the variables’ units (it is not a normalized measure).

### Coefficient of Determination (R-squared)

[R-Squared](https://www.geeksforgeeks.org/r-squared/) is a statistic that indicates how much variation the developed model can explain or capture. It is always in the range of 0 to 1. In general, the better the model matches the data, the greater the R-squared number.  
In mathematical notation, it can be expressed as:  
𝑅2=1−(𝑅𝑆𝑆𝑇𝑆𝑆)*R*2=1−(*TSSRSS*​)

* **Residual sum of Squares (RSS): The** sum of squares of the residual for each data point in the plot or data is known as the residual sum of squares, or RSS. It is a measurement of the difference between the output that was observed and what was anticipated.  
  𝑅𝑆𝑆=∑𝑖=2𝑛(𝑦𝑖−𝑏0−𝑏1𝑥𝑖)2*RSS*=∑*i*=2*n*​(*yi*​−*b*0​−*b*1​*xi*​)2
* **Total Sum of Squares (TSS):**The sum of the data points’ errors from the answer variable’s mean is known as the total sum of squares, or TSS.  
  𝑇𝑆𝑆=∑(𝑦−𝑦𝑖‾)2*TSS*=∑​(*y*−*yi*​​)2

R squared metric is a measure of the proportion of variance in the dependent variable that is explained the independent variables in the model.

### Adjusted R-Squared Error

Adjusted R2 measures the proportion of variance in the dependent variable that is explained by independent variables in a regression model. [Adjusted R-square](https://www.geeksforgeeks.org/ml-adjusted-r-square-in-regression-analysis/) accounts the number of predictors in the model and penalizes the model for including irrelevant predictors that don’t contribute significantly to explain the variance in the dependent variables.

Mathematically, adjusted R2 is expressed as:

𝐴𝑑𝑗𝑢𝑠𝑡𝑒𝑑𝑅2=1–((1−𝑅2).(𝑛−1)𝑛−𝑘−1)*AdjustedR*2=1–(*n*−*k*−1(1−*R*2).(*n*−1)​)

Here,

* n is the number of observations
* k is the number of predictors in the model
* R2 is coeeficient of determination

Adjusted R-square helps to prevent overfitting. It penalizes the model with additional predictors that do not contribute significantly to explain the variance in the dependent variable.

**8. Distinguish :**

**1. Descriptive vs. predictive models**

**Difference Between Descriptive and Predictive Data Mining:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Comparison** | **Descriptive Data Mining** | **Predictive Data Mining** |
| **1.** | **Basic** | **It determines, what happened in the past by analyzing stored data.** | **It determines, what can happen in the future with the help past data analysis.** |
| **2.** | **Preciseness** | **It provides accurate data.** | **It produces results does not ensure accuracy.** |
| **3.** | **Practical analysis methods** | **Standard reporting, query/drill down and ad-hoc reporting.** | **Predictive modelling, forecasting, simulation and alerts.** |
| **4.** | **Require** | **It requires data aggregation and data mining** | **It requires statistics and forecasting methods** |
| **5.** | **Type of approach** | **Reactive approach** | **Proactive approach** |
| **6.** | **Describe** | **Describes the characteristics of the data in a target data set.** | **Carry out the induction over the current and past data so that predictions can be made.** |
| **7.** | **Methods(in general)** | * **what happened?** * **where exactly is the problem?** * **what is the frequency of the problem?** | * **what will happen next?** * **what is the outcome if these trends continue?** * **what actions are required to be taken?** |

1. **Underfitting vs. overfitting the model**

# Overfitting and Underfitting in Machine Learning

Overfitting and Underfitting are the two main problems that occur in machine learning and degrade the performance of the machine learning models.

The main goal of each machine learning model is **to generalize well**. Here **generalization** defines the ability of an ML model to provide a suitable output by adapting the given set of unknown input. It means after providing training on the dataset, it can produce reliable and accurate output. Hence, the underfitting and overfitting are the two terms that need to be checked for the performance of the model and whether the model is generalizing well or not.

Before understanding the overfitting and underfitting, let's understand some basic term that will help to understand this topic well:

ADVERTISEMENT

ADVERTISEMENT

* **Signal:** It refers to the true underlying pattern of the data that helps the machine learning model to learn from the data.
* **Noise:** Noise is unnecessary and irrelevant data that reduces the performance of the model.
* **Bias:** Bias is a prediction error that is introduced in the model due to oversimplifying the machine learning algorithms. Or it is the difference between the predicted values and the actual values.
* **Variance:** If the machine learning model performs well with the training dataset, but does not perform well with the test dataset, then variance occurs.

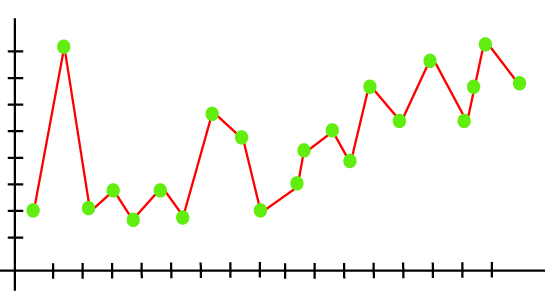
## Overfitting

Overfitting occurs when our [machine learning](https://www.javatpoint.com/machine-learning) model tries to cover all the data points or more than the required data points present in the given dataset. Because of this, the model starts caching noise and inaccurate values present in the dataset, and all these factors reduce the efficiency and accuracy of the model. The overfitted model has **low bias** and **high variance.**

The chances of occurrence of overfitting increase as much we provide training to our model. It means the more we train our model, the more chances of occurring the overfitted model.

Overfitting is the main problem that occurs in [supervised learning](https://www.javatpoint.com/supervised-machine-learning).

**Example:** The concept of the overfitting can be understood by the below graph of the linear regression output:



As we can see from the above graph, the model tries to cover all the data points present in the scatter plot. It may look efficient, but in reality, it is not so. Because the goal of the regression model to find the best fit line, but here we have not got any best fit, so, it will generate the prediction errors.

### How to avoid the Overfitting in Model

Both overfitting and underfitting cause the degraded performance of the machine learning model. But the main cause is overfitting, so there are some ways by which we can reduce the occurrence of overfitting in our model.

* **Cross-Validation**
* **Training with more data**
* **Removing features**
* **Early stopping the training**
* **Regularization**
* **Ensembling**

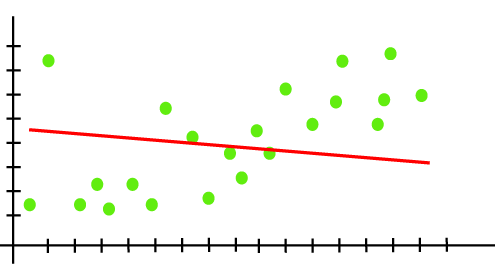
## Underfitting

Underfitting occurs when our machine learning model is not able to capture the underlying trend of the data. To avoid the overfitting in the model, the fed of training data can be stopped at an early stage, due to which the model may not learn enough from the training data. As a result, it may fail to find the best fit of the dominant trend in the data.

In the case of underfitting, the model is not able to learn enough from the training data, and hence it reduces the accuracy and produces unreliable predictions.

An underfitted model has high bias and low variance.

**Example:** We can understand the underfitting using below output of the linear regression model:



As we can see from the above diagram, the model is unable to capture the data points present in the plot.

### How to avoid underfitting:

* By increasing the training time of the model.
* By increasing the number of features.

1. **Bootstrapping vs. cross-validation**

Bootstrapping is any test or metric that relies on random sampling with replacement.It is a method that helps in many situations like validation of a predictive model performance, ensemble methods, estimation of bias and variance of the parameter of a model etc. It works by performing sampling with replacement from the original dataset, and at the same time assuming that the data points that have not been choses are the test dataset. We can repeat this procedure several times and compute the average score as estimation of our model performance. Also, Bootstrapping is related to the ensemble training methods, because we can build a model using each bootstrap datasets and “bag” these models in an ensemble using the majority voting (for classification) or computing the average (for numerical predictions) for all of these models as our final result.

Cross validation is a procedure for validating a model's performance, and it is done by splitting the training data into k parts. We assume that the k-1 parts is the training set and use the other part is our test set. We can repeat that k times differently holding out a different part of the data every time. Finally, we take the average of the k scores as our performance estimation. Cross validation can suffer from bias or variance. Increasing the number of splits, the variance will increase too and the bias will decrease. On the other hand, if we decrease the number of splits, the bias will increase and the variance will decrease.

In summary, Cross validation splits the available dataset to create multiple datasets, and Bootstrapping method uses the original dataset to create multiple datasets after resampling with replacement. Bootstrapping it is not as strong as Cross validation when it is used for model validation. Bootstrapping is more about building ensemble models or just estimating parameters.

**10. Make quick notes on:**

**1. LOOCV.**

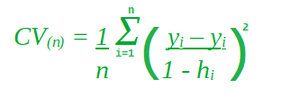
# LOOCV (Leave One Out Cross-Validation) in R Programming

**Last Updated :**20 Apr, 2023

**LOOCV(Leave One Out Cross-Validation)** is a type of [cross-validation](https://www.geeksforgeeks.org/cross-validation-in-r-programming/) approach in which each observation is considered as the validation set and the rest (N-1) observations are considered as the training set. In LOOCV, fitting of the model is done and predicting using one observation validation set. Furthermore, repeating this for N times for each observation as the validation set. Model is fitted and the model is used to predict a value for observation. This is a special case of **K-fold cross-validation** in which the number of folds is the same as the number of observations(K = N). This method helps to reduce **Bias**and **Randomness.**The method aims at reducing the Mean-Squared error rate and prevent over fitting. It is very much easy to perform LOOCV in [R programming](https://www.geeksforgeeks.org/introduction-to-r-programming-language/).

#### Mathematical Expression

LOOCV involves one fold per observation i.e each observation by itself plays the role of the validation set. The (N-1) observations play the role of the training set. With least-squares linear, a single model performance cost is the same as a single model. In LOOCV, refitting of the model can be avoided while implementing the LOOCV method. [**MSE(Mean squared error)**](https://www.geeksforgeeks.org/ml-mathematical-explanation-of-rmse-and-r-squared-error/) is calculated by fitting on the complete dataset.



In the above formula, ***hi*** represents how much influence an observation has on its own fit i.e between 0 and 1 that punishes the residual, as it divides by a small number. It inflates the residual.

#### Implementation in R

1. **F-measurement**

# F1 Score in Machine Learning

**Last Updated :**27 Dec, 2023

The F1 score is an important evaluation metric that is commonly used in classification tasks to evaluate the performance of a model. It combines precision and recall into a single value. In this article, we will understand in detail how the F1 score is calculated and compare it with other metrics.

## What is an F1 score?

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

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

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

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

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

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

F1 score combines precision and recall into a single metric

### Why harmonic mean and not simply average?

The harmonic mean is the equivalent of the arithmetic mean for reciprocals of quantities that should be averaged by the arithmetic mean. More precisely, with the harmonic mean, you transform all your numbers to the “averageable” form (by taking the reciprocal), you take their arithmetic mean and then transform the result back to the original representation (by taking the reciprocal again).

If we look at precision and recall their numerators are the same but denominators are different. So to take the average we of this quantity we need to convert them to the same base. This is done by harmonic means.

## How to calculate F1 Score?

Let us first understand confusion matrix . then we will understand how F1 score is calculated using confusion matrix for binary classification. We will then extend the concept to multi-class.

A [confusion matrix](https://www.geeksforgeeks.org/confusion-matrix-machine-learning/) is a N\*N matrix used in classification to evaluate the performance of a machine learning model. It summarizes the results of the model’s predictions on a set of data, comparing the predicted labels to the actual labels. The four components of a confusion matrix are

1. **True Positive (TP) :** The number of instances correctly predicted as positive by the model.In a binary classification problem, TP would be the number of actual positive instances that the model correctly predicted as positive.
2. **False Positive (FP) :** The number of instances incorrectly predicted as positive by the model.In a binary classification problem, FP would be the number of actual negative instances that the model incorrectly predicted as positive
3. **True Negative (TN) :** The number of instances correctly predicted as negative by the model.In a binary classification problem, TN would be the number of actual negative instances that the model correctly predicted as negative
4. **False Negative (FN) :** The number of instances incorrectly predicted as negative by the model.In a binary classification problem, FN would be the number of actual positive instances that the model incorrectly predicted as negative.

A 2\*2 matrix for binary classification can be represented as

|  | **Actual Positive** | **Actual Negative** |
| --- | --- | --- |
| **Predicted Positive** | TP | FP |
| **Predicted Negative** | FN | TN |

These components are often used to calculate various performance metrics for a classification model. The Precision, Recall , Accuracy and F1score can be calculated as below.

|  | **Actual** | | **Total** | **Formula** |
| --- | --- | --- | --- | --- |
| **Model Prediction** | TP | FP | Total Predicted Positive cases | Precision = (TP/(TP+FP)) |
| FN | TN | Total Predicted Negative cases |  |
| **Total** | Total Actual Positive Case | Total Actual Negative Case | Total Cases |  |
| **Formula** | Recall = (TP/(TP+FN)) |  | Accuracy = (TP+TN)/(TP+FP+FN+TN) | F1 = (2 \* P \* R)/(P+R) |

### Binary Classification

Let’s take an example of a dataset with 100 total cases. Out of these 90 are positive and 10 are negative cases. The model predicted 85 positive cases out of which 80 are actual positive and 5 are from actual negative cases. The confusion matrix would look like

| **Example** | **Actual** | | **Total** |  |
| --- | --- | --- | --- | --- |
| **Model Prediction** | 80 | 5 | 85 | Precision = (80/85) = 0.94 |
| 10 | 5 | 15 |  |
| **Total** | 90 | 10 | 100 |  |
|  | Recall = (80/90) = 0.88 |  | Accuracy = (80+5)/100 = 85% | F1 = 0.91 |

Let us see how does F1 score help when there is a class imbalnce

#### ****Example 1****

Consider the below case where there are only 9 cases of true positives out of a dataset of 100.

|  | **Actual** | | **Total** |
| --- | --- | --- | --- |
| **Model Prediction** | 1 | 1 | 2 |
| 8 | 90 | 98 |
| **Total** | 9 | 91 | 100 |

Precision 0.50  
Recall 0.11  
Accuracy 0.91  
F1 0.18

In this case, if we give importance to accuracy over model will predict everything as negative. This gives us an accuracy of 91 %. However, our F1 score is low

#### Example 2:

However one must also consider the opposite case where the positives outweigh the negative cases. In such a case our model will try to predict everything as positive.

|  | **Actual** |  | **Total** |
| --- | --- | --- | --- |
| **Model Prediction** | 90 | 8 | 98 |
| 1 | 1 | 2 |
| **Total** | 91 | 9 | 100 |

Precision 0.92  
Recall 0.99  
Accuracy 0.91  
F1 0.95

Here we get a good F1 score but low accuracy. In such cases, the negative should be treated as positive and positive as negative.

### Multiclass Classification

In a [multi-class classification](https://www.geeksforgeeks.org/multiclass-classification-using-scikit-learn/) problem, where there are more than two classes, we calculate the F1 score per class rather than providing a single overall F1 score for the entire model. This approach is often referred to as the one-vs-rest (OvR) or one-vs-all (OvA) strategy.

For each class in the multi-class problem, a binary classification problem is created. Essentially, we treat one class as the positive class, and the rest of the classes as the negative class. Then we proceed to calculate the F1 score as outlined above. For a specific class, the true positives (TP) are the instances correctly classified as that class, false positives (FP) are instances incorrectly classified as that class, and false negatives (FN) are instances of that class incorrectly classified as other classes.

This means that you train a separate binary classifier for each class, considering instances of that class as positive and instances of all other classes as negative.

Once we have calculated the F1 score for each class, we might want to aggregate these scores to get an overall performance measure for your model. Common approaches include calculating a micro-average, macro-average, or weighted average of the individual F1 scores.

* Micro-average involves calculating the total true positives, false positives, and false negatives across all classes and then computing precision, recall, and F1 score.
  + Micro F1 =
  + where
    - Micro Precision =
    - Micro Rcall =
* Macro-average calculates the average of F1 scores for each class without considering class imbalance.
  + Macro F1 =
  + where
    - N is the number of classes
    - F1i is F1 score for ith class
* Weighted average considers class imbalance and weights the F1 scores by the number of instances in each class.
  + Weighted F1 =
  + where
    - Weighti =

## F1 Score vs ROC-AUC vs Accuracy

Besides the F1 score, there are other metrics like accuracy, [AUC-ROC](https://www.geeksforgeeks.org/auc-roc-curve/), etc which can be used to evaluate model performance. The choice of metric depends on the problem at hand. There is no one-size-fits-all all. More than often a combination of metrics are looked at to gauge the overall performance of the model. Below are general rules that are followed :

### F1 vs Accuracy

If the problem is balanced and you care about both positive and negative predictions, accuracy is a good choice. If the problem is imbalanced(a lot of negative cases compared to positive) and we need to focus on positive cases the F1 score is a good choice.

### F1 vs AUC-ROC

AUC-ROC helps us to understand the ability of the model to discriminate between positive and negative instances overall, regardless of class imbalance at different thresholds while the F1 score evaluates the performance of the model at a particular threshold. Hence one might use F1 for class-specific evaluation while AUC-ROC for overall assessment of model.

1. **The width of the silhouette**

# Silhouette Algorithm to determine the optimal value of k

**Last Updated :**06 Jun, 2019

One of the fundamental steps of an unsupervised learning algorithm is to determine the number of clusters into which the data may be divided. The silhouette algorithm is one of the many algorithms to determine the optimal number of clusters for an unsupervised learning technique. In the Silhouette algorithm, we assume that the data has already been clustered into k clusters by a clustering technique(Typically [K-Means Clustering technique](https://www.geeksforgeeks.org/k-means-clustering-introduction/)). Then for each data point, we define the following:- C(i) -The cluster assigned to the ith data point |C(i)| – The number of data points in the cluster assigned to the ith data point a(i) – It gives a measure of how well assigned the ith data point is to it’s clusterb(i) – It is defined as the average dissimilarity to the closest cluster which is not it’s clusterThe silhouette coefficient s(i) is given by:-We determine the average silhouette for each value of k and for the value of k which has the **maximum value of s(i)** is considered the optimal number of clusters for the unsupervised learning algorithm. Let us consider the following data:-

| **S.No** | **X1** | **X2** |
| --- | --- | --- |
| 1. | -7.36 | 6.37 |
| 2. | 3.08 | -6.78 |
| 3. | 5.03 | -8.31 |
| 4. | -1.93 | -0.92 |
| 5. | -8.86 | 6.60 |

We now iterate the values of k from 2 to 5. We assume that no practical data exists for which all the data points can be optimally clustered into 1 cluster. We construct the following tables for each value of k:- **k = 2**

| **S.No** | **a(i)** | **b(i)** | **s(i)** |
| --- | --- | --- | --- |
| 1. | 5.31 | 14.1 | 0.62 |
| 2. | 2.47 | 13.15 | 0.81 |
| 3. | 2.47 | 14.97 | 0.84 |
| 4. | 9.66 | 8.93 | -0.076 |
| 5. | 5.88 | 19.16 | 0.69 |

**Average value of s(i) = 0.58** **k = 3**

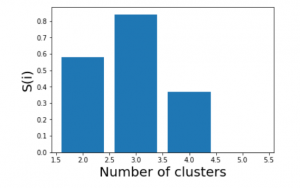
| **S.No** | **a(i)** | **b(i)** | **s(i)** |
| --- | --- | --- | --- |
| 1. | 1.52 | 9.09 | 0.83 |
| 2. | 2.47 | 7.71 | 0.68 |
| 3. | 2.47 | 10.15 | 0.76 |
| 4. | 0 | 7.71 | 1 |
| 5. | 1.52 | 17.93 | 0.92 |

**Average value of s(i) = 0.84** **k = 4**

| **S.No** | **a(i)** | **b(i)** | **s(i)** |
| --- | --- | --- | --- |
| 1. | 1.52 | 9.09 | 0.83 |
| 2. | infinite | 2.47 | 0 |
| 3. | infinite | 2.47 | 0 |
| 4. | infinite | 7.71 | 0 |
| 5. | 1.52 | 10.23 | 0.85 |

**Average value of s(i) = 0.37** **k = 5**

| **S.No** | **a(i)** | **b(i)** | **s(i)** |
| --- | --- | --- | --- |
| 1. | infinite | 1.52 | 0 |
| 2. | infinite | 2.47 | 0 |
| 3. | infinite | 2.47 | 0 |
| 4. | infinite | 7.71 | 0 |
| 5. | infinite | 1.52 | 0 |

**Average value of s(i) = 0**We see that the highest value of s(i) exists for k = 3. Therefore we conclude that the optimal number of clusters for the given data is 3.

**4. Receiver operating characteristic curve**

# AUC ROC Curve in Machine Learning

**Last Updated :**25 Jan, 2024

One important aspect of [Machine Learning](https://www.geeksforgeeks.org/machine-learning/) is model evaluation. You need to have some mechanism to evaluate your model. This is where these performance metrics come into the picture they give us a sense of how good a model is. If you are familiar with some of the basics of [Machine Learning](https://www.geeksforgeeks.org/machine-learning/) then you must have come across some of these metrics, like accuracy, precision, recall, auc-roc, etc., which are generally used for classification tasks. In this article, we will explore in depth one such metric, which is the AUC-ROC curve.

**Table of Content**

* [What is the AUC-ROC curve?](https://www.geeksforgeeks.org/auc-roc-curve/#what-is-the-aucroc-curve)
* [Key terms used in AUC and ROC Curve](https://www.geeksforgeeks.org/auc-roc-curve/#key-terms-used-in-auc-and-roc-curve)
* [Relationship between Sensitivity, Specificity, FPR, and Threshold.](https://www.geeksforgeeks.org/auc-roc-curve/#relationship-between-sensitivity-specificity-fpr-and-threshold)
* [How does AUC-ROC work?](https://www.geeksforgeeks.org/auc-roc-curve/#how-does-aucroc-work)
* [When should we use the AUC-ROC evaluation metric?](https://www.geeksforgeeks.org/auc-roc-curve/#when-should-we-use-the-aucroc-evaluation-metric)
* [Speculating the performance of the model](https://www.geeksforgeeks.org/auc-roc-curve/#speculating-the-performance-of-the-model)
* [Understanding the AUC-ROC Curve](https://www.geeksforgeeks.org/auc-roc-curve/#understanding-the-aucroc-curve)
* [Implementation using two different models](https://www.geeksforgeeks.org/auc-roc-curve/#implementation-using-two-different-models)
* [How to use ROC-AUC for a multi-class model?](https://www.geeksforgeeks.org/auc-roc-curve/#how-to-use-rocauc-for-a-multiclass-model)
* [FAQs for AUC ROC Curve in Machine Learning](https://www.geeksforgeeks.org/auc-roc-curve/#faqs-for-auc-roc-curve-in-machine-learning)

## What is the AUC-ROC curve?

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

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

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

### ****Receiver Operating Characteristics (ROC) Curve****

ROC stands for Receiver Operating Characteristics, and the ROC curve is the graphical representation of the effectiveness of the binary classification model. It plots the true positive rate (TPR) vs the false positive rate (FPR) at different classification thresholds.

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

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

 It represents the [probability](https://www.geeksforgeeks.org/probability-gq/) with which our model can distinguish between the two classes present in our target.

*ROC-AUC Classification Evaluation Metric*

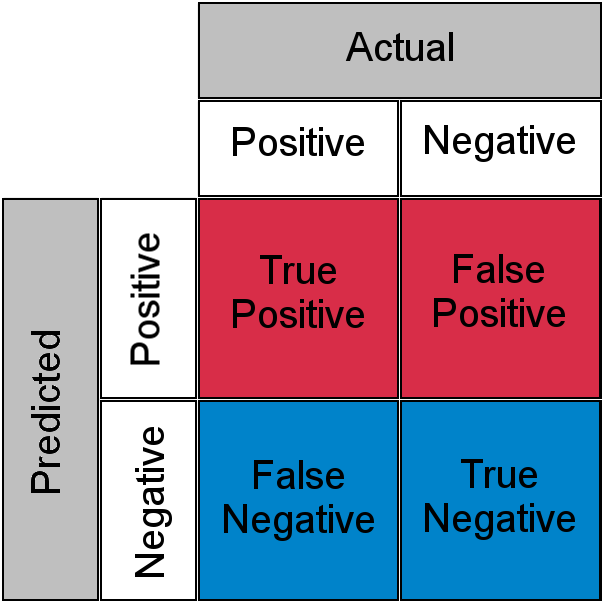
## Key terms used in AUC and ROC Curve

### 1. TPR and FPR

This is the most common definition that you would have encountered when you would Google AUC-ROC. Basically, the ROC curve is a graph that shows the performance of a classification model at all possible thresholds( threshold is a particular value beyond which you say a point belongs to a particular class). The curve is plotted between two parameters

* **TPR** – True Positive Rate
* **FPR** – False Positive Rate

Before understanding, TPR and FPR let us quickly look at the [confusion matrix](https://www.geeksforgeeks.org/confusion-matrix-machine-learning/).



*Confusion Matrix for a Classification Task*

* **True Positive**: Actual Positive and Predicted as Positive
* **True Negative**: Actual Negative and Predicted as Negative
* **False Positive(Type I Error)**: Actual Negative but predicted as Positive
* **False Negative(Type II Error)**: Actual Positive but predicted as Negative

In simple terms, you can call False Positive a **false alarm** and False Negative a **miss**. Now let us look at what TPR and FPR are.

### 2. Sensitivity / True Positive Rate / Recall

Basically, TPR/Recall/Sensitivity is the ratio of positive examples that are correctly identified.  It represents the ability of the model to correctly identify positive instances and is calculated as follows:

Sensitivity/Recall/TPR measures the proportion of actual positive instances that are correctly identified by the model as positive.

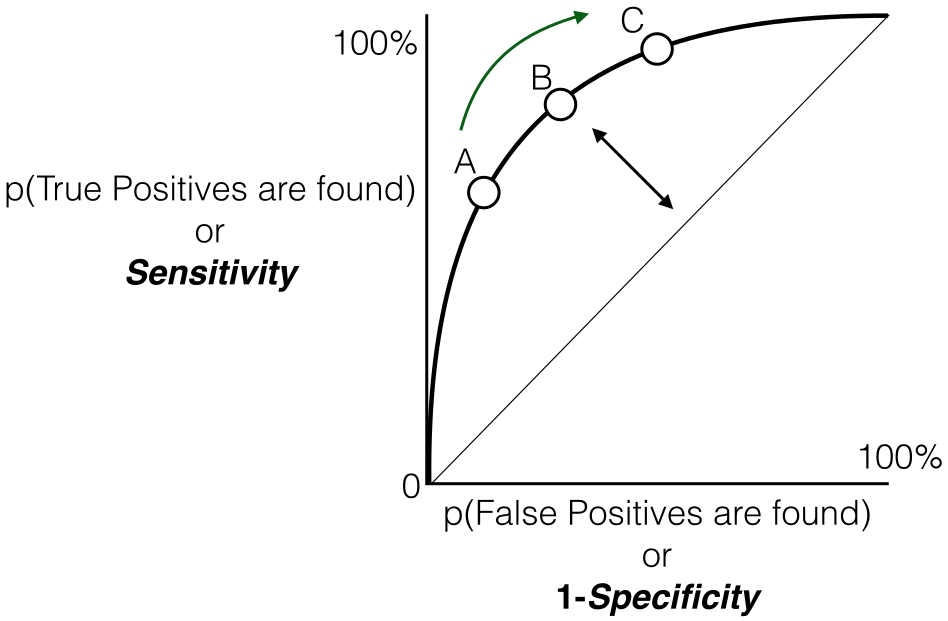
### 3. False Positive Rate

FPR is the ratio of negative examples that are incorrectly classified.

### 4. Specificity

Specificity measures the proportion of actual negative instances that are correctly identified by the model as negative. It represents the ability of the model to correctly identify negative instances

And as said earlier ROC is nothing but the plot between TPR and FPR across all possible thresholds and AUC is the entire area beneath this ROC curve.



*Sensitivity versus False Positive Rate plot*

## Relationship between Sensitivity, Specificity, FPR, and Threshold****.****

### ****Sensitivity and Specificity:****

* **Inverse Relationship:**  sensitivity and specificity have an inverse relationship. When one increases, the other tends to decrease. This reflects the inherent trade-off between true positive and true negative rates.
* **Tuning via Threshold:** By adjusting the threshold value, we can control the balance between sensitivity and specificity. Lower thresholds lead to higher sensitivity (more true positives) at the expense of specificity (more false positives). Conversely, raising the threshold boosts specificity (fewer false positives) but sacrifices sensitivity (more false negatives).

### ****Threshold and False Positive Rate (FPR):****

* **FPR and Specificity Connection:** False Positive Rate (FPR) is simply the complement of specificity (FPR = 1 – specificity). This signifies the direct relationship between them: higher specificity translates to lower FPR, and vice versa.
* **FPR Changes with TPR:** Similarly, as you observed, the True Positive Rate (TPR) and FPR are also linked. An increase in TPR (more true positives) generally leads to a rise in FPR (more false positives). Conversely, a drop in TPR (fewer true positives) results in a decline in FPR (fewer false positives)

## How does AUC-ROC work?

We looked at the geometric interpretation, but I guess it is still not enough in developing the intuition behind what 0.75 AUC actually means, now let us look at AUC-ROC from a probabilistic point of view. Let us first talk about what AUC does and later we will build our understanding on top of this

***AUC measures how well a model is able to distinguish between classes.***

An AUC of 0.75 would actually mean that let’s say we take two data points belonging to separate classes then there is a 75% chance the model would be able to segregate them or rank order them correctly i.e positive point has a higher prediction probability than the negative class. (assuming a higher prediction probability means the point would ideally belong to the positive class). Here is a small example to make things more clear.

| **Index** | **Class** | **Probability** |
| --- | --- | --- |
| **P1** | 1 | 0.95 |
| **P2** | 1 | 0.90 |
| **P3** | 0 | 0.85 |
| **P4** | 0 | 0.81 |
| **P5** | 1 | 0.78 |
| **P6** | 0 | 0.70 |

Here we have 6 points where P1, P2, and P5 belong to class 1 and P3, P4, and P6 belong to class 0 and we’re corresponding predicted probabilities in the Probability column, as we said if we take two points belonging to separate classes then what is the probability that model rank orders them correctly.

We will take all possible pairs such that one point belongs to class 1 and the other belongs to class 0, we will have a total of 9 such pairs below are all of these 9 possible pairs.

| **Pair** | **isCorrect** |
| --- | --- |
| **(P1,P3)** | Yes |
| **(P1,P4)** | Yes |
| **(P1,P6)** | Yes |
| **(P2,P3)** | Yes |
| **(P2,P4)** | Yes |
| **(P2,P6)** | Yes |
| **(P3,P5)** | No |
| **(P4,P5)** | No |
| **(P5,P6)** | Yes |

Here column is Correct tells if the mentioned pair is correctly rank-ordered based on the predicted probability i.e class 1 point has a higher probability than class 0 point, in 7 out of these 9 possible pairs class 1 is ranked higher than class 0, or we can say that there is a 77% chance that if you pick a pair of points belonging to separate classes the model would be able to distinguish them correctly. Now, I think you might have a bit of intuition behind this AUC number, just to clear up any further doubts let’s validate it using Scikit learns AUC-ROC implementation.

* Python3

|  |
| --- |
| import numpy as np  from sklearn .metrics import roc\_auc\_score    y\_true = [1, 1, 0, 0, 1, 0]  y\_pred = [0.95, 0.90, 0.85, 0.81, 0.78, 0.70]  auc = np.round(roc\_auc\_score(y\_true, y\_pred), 3)  print("Auc for our sample data is {}".format(auc)) |

**Output:**

AUC for our sample data is 0.778

## When should we use the AUC-ROC evaluation metric?

There are some areas where using ROC-AUC might not be ideal. In cases where the dataset is highly imbalanced, **the ROC curve can give an overly optimistic assessment of the model’s performance**. This optimism bias arises because the ROC curve’s false positive rate (FPR) can become very small when the number of actual negatives is large.

Looking at the FPR formula,

**we observe**,

* The Negative class is in the majority, the denominator of FPR is dominated by True Negatives, because of which FPR becomes less sensitive to changes in predictions related to the minority class (positive class).
* ROC curves may be appropriate when the cost of False Positives and False Negatives is balanced and the dataset is not heavily imbalanced.

In those case, [Precision-Recall Curves](https://www.geeksforgeeks.org/precision-recall-curve-ml/)can be used which provide an alternative evaluation metric that is more suitable for imbalanced datasets, focusing on the performance of the classifier with respect to the positive (minority) class.

## Speculating the performance of the model

* A high AUC (close to 1) indicates excellent discriminative power. This means the model is effective in distinguishing between the two classes, and its predictions are reliable.
* A low AUC (close to 0) suggests poor performance. In this case, the model struggles to differentiate between the positive and negative classes, and its predictions may not be trustworthy.
* AUC around 0.5 implies that the model is essentially making random guesses. It shows no ability to separate the classes, indicating that the model is not learning any meaningful patterns from the data.

## Understanding the AUC-ROC Curve

In an ROC curve, the x-axis typically represents the False Positive Rate (FPR), and the y-axis represents the True Positive Rate (TPR), also known as Sensitivity or Recall. So, a higher x-axis value (towards the right) on the ROC curve does indicate a higher False Positive Rate, and a higher y-axis value (towards the top) indicates a higher True Positive Rate.The ROC curve is a graphical representation of the trade-off between true positive rate and false positive rate at various thresholds. It shows the performance of a classification model at different classification thresholds. The AUC (Area Under the Curve) is a summary measure of the ROC curve performance.The choice of the threshold depends on the specific requirements of the problem you’re trying to solve and the trade-off between false positives and false negatives that is acceptable in your context.

* If you want to prioritize reducing false positives (minimizing the chances of labeling something as positive when it’s not), you might choose a threshold that results in a lower false positive rate.
* If you want to prioritize increasing true positives (capturing as many actual positives as possible), you might choose a threshold that results in a higher true positive rate.

Let’s consider an example to illustrate how ROC curves are generated for different [thresholds](https://www.geeksforgeeks.org/decision-threshold-in-machine-learning/)and how a particular threshold corresponds to a confusion matrix. Suppose we have a [binary classification problem](https://www.geeksforgeeks.org/basic-concept-classification-data-mining/) with a model predicting whether an email is spam (positive) or not spam (negative).

Let us consider the hypothetical data,

True Labels: [1, 0, 1, 0, 1, 1, 0, 0, 1, 0]

Predicted Probabilities: [0.8, 0.3, 0.6, 0.2, 0.7, 0.9, 0.4, 0.1, 0.75, 0.55]

### ****Case 1: Threshold = 0.5****

| **True Labels** | **Predicted Probabilities** | **Predicted Labels (if Threshold = 0.5)** |
| --- | --- | --- |
| 1 | 0.8 | 1 |
| 0 | 0.3 | 0 |
| 1 | 0.6 | 1 |
| 0 | 0.2 | 0 |
| 1 | 0.7 | 1 |
| 1 | 0.9 | 1 |
| 0 | 0.4 | 0 |
| 0 | 0.1 | 0 |
| 1 | 0.75 | 1 |
| 0 | 0.55 | 1 |

#### Confusion matrix based on above predictions

|  | **Prediction = 0** | **Prediction = 1** |
| --- | --- | --- |
| **Actual = 0** | TP=4 | FN=1 |
| **Actual = 1** | FP=0 | TN=5 |

Accordingly,

* **True Positive Rate (TPR)**:  
  Proportion of actual positives correctly identified by the classifier is
* **False Positive Rate (FPR)**:  
  Proportion of actual negatives incorrectly classified as positives

So, at the threshold of 0.5:

* True Positive Rate (Sensitivity): 0.8
* False Positive Rate: 0

The interpretation is that the model, at this threshold, correctly identifies 80% of actual positives (TPR) but incorrectly classifies 0% of actual negatives as positives (FPR).

Accordingly for different thresholds we will get ,

### ****Case 2: Threshold = 0.7****

| **True Labels** | **Predicted Probabilities** | **Predicted Labels (if Threshold = 0.7)** |
| --- | --- | --- |
| 1 | 0.8 | 1 |
| 0 | 0.3 | 0 |
| 1 | 0.6 | 0 |
| 0 | 0.2 | 0 |
| 1 | 0.7 | 0 |
| 1 | 0.9 | 1 |
| 0 | 0.4 | 0 |
| 0 | 0.1 | 0 |
| 1 | 0.75 | 1 |
| 0 | 0.55 | 0 |

#### Confusion matrix based on above predictions

|  | **Prediction = 0** | **Prediction = 1** |
| --- | --- | --- |
| **Actual = 0** | TP=5 | FN=0 |
| **Actual = 1** | FP=2 | TN=3 |

Accordingly,

* **True Positive Rate (TPR)**:  
  Proportion of actual positives correctly identified by the classifier is
* **False Positive Rate (FPR)**:  
  Proportion of actual negatives incorrectly classified as positives

### ****Case 3: Threshold = 0.4****

| **True Labels** | **Predicted Probabilities** | **Predicted Labels (if Threshold = 0.4)** |
| --- | --- | --- |
| 1 | 0.8 | 1 |
| 0 | 0.3 | 0 |
| 1 | 0.6 | 1 |
| 0 | 0.2 | 0 |
| 1 | 0.7 | 1 |
| 1 | 0.9 | 1 |
| 0 | 0.4 | 0 |
| 0 | 0.1 | 0 |
| 1 | 0.75 | 1 |
| 0 | 0.55 | 1 |

#### Confusion matrix based on above predictions

|  | **Prediction = 0** | **Prediction = 1** |
| --- | --- | --- |
| **Actual = 0** | TP=4 | FN=1 |
| **Actual = 1** | FP=0 | TN=5 |

Accordingly,

* **True Positive Rate (TPR)**:  
  Proportion of actual positives correctly identified by the classifier is
* **False Positive Rate (FPR)**:  
  Proportion of actual negatives incorrectly classified as positives

### ****Case 4: Threshold = 0.2****

| **True Labels** | **Predicted Probabilities** | **Predicted Labels (if Threshold = 0.2)** |
| --- | --- | --- |
| 1 | 0.8 | 1 |
| 0 | 0.3 | 1 |
| 1 | 0.6 | 1 |
| 0 | 0.2 | 0 |
| 1 | 0.7 | 1 |
| 1 | 0.9 | 1 |
| 0 | 0.4 | 1 |
| 0 | 0.1 | 0 |
| 1 | 0.75 | 1 |
| 0 | 0.55 | 1 |

#### Confusion matrix based on above predictions

|  | **Prediction = 0** | **Prediction = 1** |
| --- | --- | --- |
| **Actual = 0** | TP=2 | FN=3 |
| **Actual = 1** | FP=0 | TN=5 |

Accordingly,

* **True Positive Rate (TPR)**:  
  Proportion of actual positives correctly identified by the classifier is
* **False Positive Rate (FPR)**:  
  Proportion of actual negatives incorrectly classified as positives

### ****Case 5: Threshold = 0.85****

| **True Labels** | **Predicted Probabilities** | **Predicted Labels (if Threshold = 0.85)** |
| --- | --- | --- |
| 1 | 0.8 | 0 |
| 0 | 0.3 | 0 |
| 1 | 0.6 | 0 |
| 0 | 0.2 | 0 |
| 1 | 0.7 | 0 |
| 1 | 0.9 | 1 |
| 0 | 0.4 | 0 |
| 0 | 0.1 | 0 |
| 1 | 0.75 | 0 |
| 0 | 0.55 | 0 |

#### Confusion matrix based on above predictions

|  | **Prediction = 0** | **Prediction = 1** |
| --- | --- | --- |
| **Actual = 0** | TP=5 | FN=0 |
| **Actual = 1** | FP=4 | TN=1 |

Accordingly,

* **True Positive Rate (TPR)**:  
  Proportion of actual positives correctly identified by the classifier is
* **False Positive Rate (FPR)**:  
  Proportion of actual negatives incorrectly classified as positives

Based on the above result, we will plot the ROC curve

* Python3

|  |
| --- |
| true\_positive\_rate = [0.4, 0.8,  0.8, 1.0, 1]  false\_positive\_rate = [0, 0,  0, 0.2, 0.8]    plt.plot(false\_positive\_rate, true\_positive\_rate, 'o-', label='ROC')  plt.plot([0, 1], [0, 1], '--', color='grey', label='Worst case')  plt.xlabel('False Positive Rate')  plt.ylabel('True Positive Rate')  plt.title('ROC Curve')  plt.legend()  plt.show() |

**Output:**

From the graph it is implied that:

* The gray dashed line represents the “Worst case” scenario, where the model’s predictions i.e TPR are FPR are same. This diagonal line is considered the worst-case scenario, indicating an equal likelihood of false positives and false negatives.
* As points deviate from the random guess line towards the upper-left corner, the model’s performance improves.
* The Area Under the Curve (AUC) is a quantitative measure of the model’s discriminative ability. A higher AUC value, closer to 1.0, indicates superior performance. The best possible AUC value is 1.0, corresponding to a model that achieves 100% sensitivity and 100% specificity.

In all, the Receiver Operating Characteristic (ROC) curve serves as a graphical representation of the trade-off between a binary classification model’s True Positive Rate (sensitivity) and False Positive Rate at various decision thresholds. As the curve gracefully ascends towards the upper-left corner, it signifies the model’s commendable ability to discriminate between positive and negative instances across a range of confidence thresholds. This upward trajectory indicates an improved performance, with higher sensitivity achieved while minimizing false positives. The annotated thresholds, denoted as A, B, C, D, and E, offer valuable insights into the model’s dynamic behavior at different confidence levels.

## Implementation using two different models

#### Installing Libraries

* Python3

|  |
| --- |
| import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  from sklearn.datasets import make\_classification  from sklearn.model\_selection import train\_test\_split  from sklearn.linear\_model import LogisticRegression  from sklearn.ensemble import RandomForestClassifier  from sklearn.metrics import roc\_curve, auc |

In order to train the[Random Forest](https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/) and [Logistic Regression](https://www.geeksforgeeks.org/understanding-logistic-regression/) models and to present their ROC curves with AUC scores, the algorithm creates artificial binary classification data.

#### Generating data and splitting data

* Python3

|  |
| --- |
| # Generate synthetic data for demonstration  X, y = make\_classification(      n\_samples=1000, n\_features=20, n\_classes=2, random\_state=42)    # Split the data into training and testing sets  X\_train, X\_test, y\_train, y\_test = train\_test\_split(      X, y, test\_size=0.2, random\_state=42) |

Using an 80-20 split ratio, the algorithm creates artificial binary classification data with 20 features, divides it into training and testing sets, and assigns a random seed to ensure reproducibility.

#### Training the different models

* Python3

|  |
| --- |
| # Train two different models  logistic\_model = LogisticRegression(random\_state=42)  logistic\_model.fit(X\_train, y\_train)    random\_forest\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)  random\_forest\_model.fit(X\_train, y\_train) |

Using a fixed random seed to ensure repeatability, the method initializes and trains a logistic regression model on the training set. In a similar manner, it uses the training data and the same random seed to initialize and train a Random Forest model with 100 trees.

#### Predictions

* Python3

|  |
| --- |
| # Generate predictions  y\_pred\_logistic = logistic\_model.predict\_proba(X\_test)[:, 1]  y\_pred\_rf = random\_forest\_model.predict\_proba(X\_test)[:, 1] |

Using the test data and a trained [Logistic Regression](https://www.geeksforgeeks.org/ml-logistic-regression-using-python/) model, the code predicts the positive class’s probability. In a similar manner, using the test data, it uses the trained Random Forest model to produce projected probabilities for the positive class.

#### Creating a dataframe

* Python3

|  |
| --- |
| # Create a DataFrame  test\_df = pd.DataFrame(      {'True': y\_test, 'Logistic': y\_pred\_logistic, 'RandomForest': y\_pred\_rf}) |

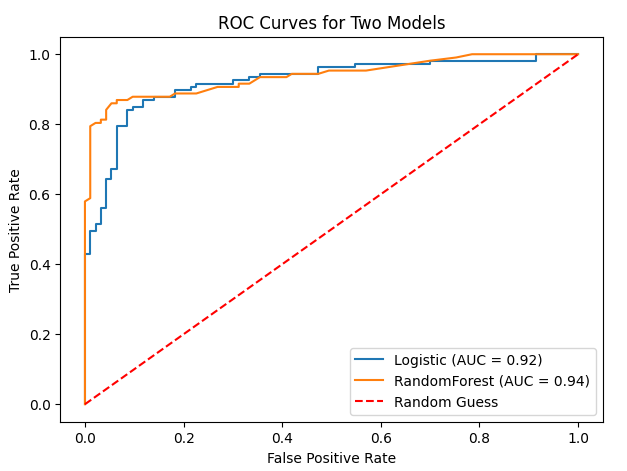
Using the test data, the code creates a DataFrame called test\_df with columns labeled “True,” “Logistic,” and “RandomForest,” adding true labels and predicted probabilities from the Random Forest and Logistic Regression models.

#### Plot the ROC Curve for the models

* Python3

|  |
| --- |
| # Plot ROC curve for each model  plt.figure(figsize=(7, 5))    for model in ['Logistic', 'RandomForest']:      fpr, tpr, \_ = roc\_curve(test\_df['True'], test\_df[model])      roc\_auc = auc(fpr, tpr)      plt.plot(fpr, tpr, label=f'{model} (AUC = {roc\_auc:.2f})')    # Plot random guess line  plt.plot([0, 1], [0, 1], 'r--', label='Random Guess')    # Set labels and title  plt.xlabel('False Positive Rate')  plt.ylabel('True Positive Rate')  plt.title('ROC Curves for Two Models')  plt.legend()  plt.show() |

**Output:**



The code generates a plot with 8 by 6 inch figures. It computes the AUC and ROC curve for each model (Random Forest and Logistic Regression), then plots the ROC curve. The [ROC curve](https://www.geeksforgeeks.org/calculate-roc-auc-for-classification-algorithm-such-as-random-forest/) for random guessing is also represented by a red dashed line, and labels, a title, and a legend are set for visualization.

## How to use ROC-AUC for a multi-class model?

For a multi-class setting, we can simply use one vs all methodology and you will have one ROC curve for each class. Let’s say you have four classes A, B, C, and D then there would be ROC curves and corresponding AUC values for all the four classes, i.e. once A would be one class and B, C, and D combined would be the others class, similarly, B is one class and A, C, and D combined as others class, etc.

The general steps for using AUC-ROC in the context of a multiclass classification model are:

#### ****One-vs-All Methodology:****

* For each class in your multiclass problem, treat it as the positive class while combining all other classes into the negative class.
* Train the binary classifier for each class against the rest of the classes.

#### Calculate AUC-ROC for Each Class:

* Here we plot the ROC curve for the given class against the rest.
* Plot the ROC curves for each class on the same graph. Each curve represents the discrimination performance of the model for a specific class.
* Examine the AUC scores for each class. A higher AUC score indicates better discrimination for that particular class.

### Implementation of AUC-ROC in Multiclass Classification

#### Importing Libraries

* Python3

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  from sklearn.datasets import make\_classification  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import label\_binarize  from sklearn.multiclass import OneVsRestClassifier  from sklearn.linear\_model import LogisticRegression  from sklearn.ensemble import RandomForestClassifier  from sklearn.metrics import roc\_curve, auc  from itertools import cycle |

The program creates artificial multiclass data, divides it into training and testing sets, and then uses the [One-vs-Restclassifier](https://www.geeksforgeeks.org/one-vs-rest-strategy-for-multi-class-classification/)technique to train classifiers for both Random Forest and Logistic Regression. Lastly, it plots the two models’ multiclass ROC curves to demonstrate how well they discriminate between various classes.

#### Generating Data and splitting

* Python3

|  |
| --- |
| # Generate synthetic multiclass data  X, y = make\_classification(      n\_samples=1000, n\_features=20, n\_classes=3, n\_informative=10, random\_state=42)    # Binarize the labels  y\_bin = label\_binarize(y, classes=np.unique(y))    # Split the data into training and testing sets  X\_train, X\_test, y\_train, y\_test = train\_test\_split(      X, y\_bin, test\_size=0.2, random\_state=42) |

Three classes and twenty features make up the synthetic multiclass data produced by the code. After label binarization, the data is divided into training and testing sets in an 80-20 ratio.

#### Training Models

* Python3

|  |
| --- |
| # Train two different multiclass models  logistic\_model = OneVsRestClassifier(LogisticRegression(random\_state=42))  logistic\_model.fit(X\_train, y\_train)    rf\_model = OneVsRestClassifier(      RandomForestClassifier(n\_estimators=100, random\_state=42))  rf\_model.fit(X\_train, y\_train) |

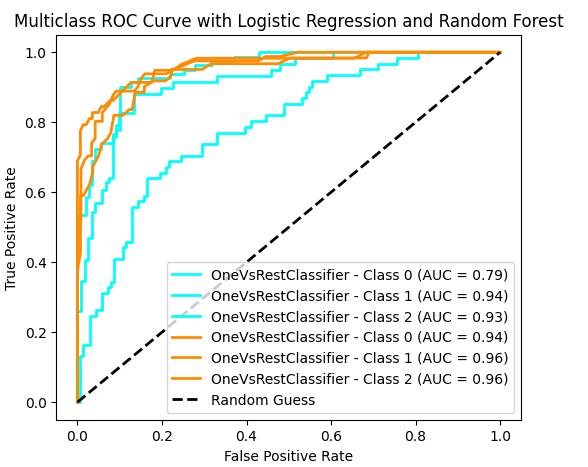
The program trains two multiclass models: a Random Forest model with 100 estimators and a Logistic Regression model with the [One-vs-Rest approach](https://www.geeksforgeeks.org/plot-multinomial-and-one-vs-rest-logistic-regression-in-scikit-learn/). With the training set of data, both models are fitted.

#### Plotting the AUC-ROC Curve

* Python3

|  |
| --- |
| # Compute ROC curve and ROC area for each class  fpr = dict()  tpr = dict()  roc\_auc = dict()    models = [logistic\_model, rf\_model]    plt.figure(figsize=(6, 5))  colors = cycle(['aqua', 'darkorange'])    for model, color in zip(models, colors):      for i in range(model.classes\_.shape[0]):          fpr[i], tpr[i], \_ = roc\_curve(              y\_test[:, i], model.predict\_proba(X\_test)[:, i])          roc\_auc[i] = auc(fpr[i], tpr[i])          plt.plot(fpr[i], tpr[i], color=color, lw=2,                   label=f'{model.\_\_class\_\_.\_\_name\_\_} - Class {i} (AUC = {roc\_auc[i]:.2f})')    # Plot random guess line  plt.plot([0, 1], [0, 1], 'k--', lw=2, label='Random Guess')    # Set labels and title  plt.xlabel('False Positive Rate')  plt.ylabel('True Positive Rate')  plt.title('Multiclass ROC Curve with Logistic Regression and Random Forest')  plt.legend(loc="lower right")  plt.show() |

**Output:**



The Random Forest and Logistic Regression models’ ROC curves and AUC scores are calculated by the code for each class. The multiclass ROC curves are then plotted, showing the discrimination performance of each class and featuring a line that represents random guessing. The resulting plot offers a graphic evaluation of the models’ classification performance.

## Conclusion

In machine learning, the performance of binary classification models is assessed using a crucial metric called the Area Under the Receiver Operating Characteristic (AUC-ROC). Across various decision thresholds, it shows how sensitivity and specificity are traded off. Greater discrimination between positive and negative instances is typically exhibited by a model with a higher AUC score. Whereas 0.5 denotes chance, 1 represents flawless performance. Model optimization and selection are aided by the useful information that the AUC-ROC curve offers about a model’s capacity to discriminate between classes. When working with unbalanced datasets or applications where false positives and false negatives have different costs, it is particularly useful as a comprehensive measure.

## FAQs for AUC ROC Curve in Machine Learning

**1. What is the AUC-ROC curve?**

*For various classification thresholds, the trade-off between true positive rate (sensitivity) and false positive rate (specificity) is graphically represented by the AUC-ROC curve.*

**2. What does a perfect AUC-ROC curve look like?**

*An area of 1 on an ideal AUC-ROC curve would mean that the model achieves optimal sensitivity and specificity at all thresholds.*

**3. What does an AUC value of 0.5 signify?**

*AUC of 0.5 indicates that the model’s performance is comparable to that of random chance. It suggests a lack of discriminating ability.*

**4. Can AUC-ROC be used for multiclass classification?**

*AUC-ROC is frequently applied to issues involving binary classification. Variations such as the macro-average or micro-average AUC can be taken into consideration for multiclass classification.*

**5. How is the AUC-ROC curve useful in model evaluation?**

*The ability of a model to discriminate between classes is comprehensively summarized by the AUC-ROC curve. When working with unbalanced datasets, it is especially helpful.*