AI Sciences

Learn Machine Learning In One Day

AI SCIENCES

AI Sciences

LEARNING MACHINE LEARNING IN ONE DAY

Introduction to Machine Learning Fundamentals for Beginners

AI Sciences Publishing

How to contact us

Please address comments and questions concerning this book to our customer service by email at:

contact@aisciences.net

Our goal is to provide high-quality books for your technical learning in data science and artificial intelligence subjects.

Thank you so much for buying this book.

If you notice any problem, please let us know by sending us an email to review@aisciences.net before writing any review online. This will help us to improve the quality of our books.

Table of Contents

First Printing, 2019

Edited by Davies Company
Ebook Converted and Cover by Pixels Studio
Publised by AI Sciences LLC

ISBN-13: 978-1-7335706-9-5
ISBN-10: 1-7335706-9-1

Legal Notice:

Disclaimer Notice:

- Do you want to discover, learn, and understand the methods and techniques of artificial intelligence, data science, computer science, machine learning, deep learning or statistics?
- Would you like to have books that you can read quickly and understand very easily?
- Would you like to practice AI techniques?

If you answered yes to any of the above, you are in the right place. The AI Sciences book series is perfectly suited to your expectations!

Our books are the best on the market for beginners, newcomers, students, and anyone who wants to learn more about these subjects without going into too much theoretical and mathematical detail. Our books are among the best sellers on Amazon in the field in general.

About Us

We are a group of experts, PhD students, and young practitioners of artificial intelligence, computer science, machine learning, and statistics. Some of us work for big-name companies like Google, Facebook, Microsoft, KPMG, BCG, and Mazars.

We decided to produce a series of books mainly dedicated to beginners and newcomers on the techniques and methods of machine learning, statistics, artificial intelligence, and data science. Initially, our objective was to help only those who wish to understand these techniques more easily and to be able to

start without too much theory or lengthy reading. Today, we also publish more complete books on selected topics for a wider audience.

About our Books

Our books have had phenomenal success and they are currently among the best sellers on Amazon. Our books have helped many people to progress and grasp these techniques, which are considered by some to be complicated.

The books we produce are short, accessible, and enjoyable. These books focus on the essentials so that beginners can quickly understand and practice the relevant techniques effectively. You will never regret having chosen one of our books.

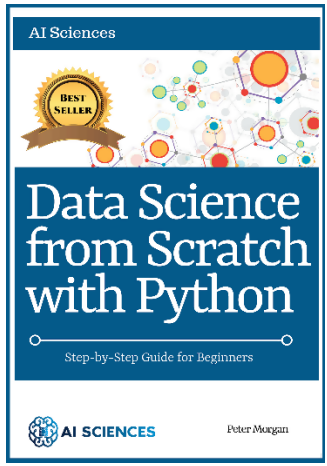
We also offer a selection of complimentary books on our website, completely free of charge: visit our site and subscribe to our mailing list for free: www.aisciences.net.

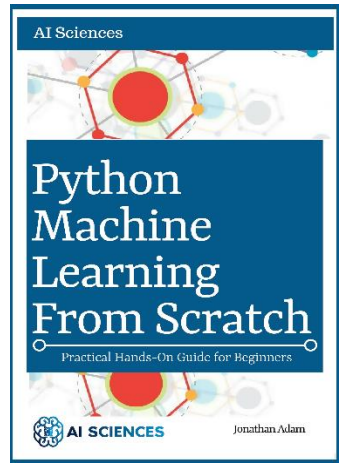
By subscribing to our mailing list, you will receive unlimited and continuous access to any of our newly published books for free.

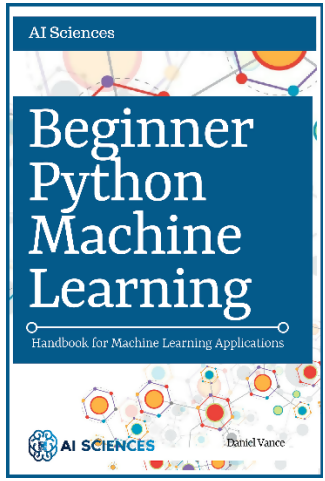
To Contact Us:

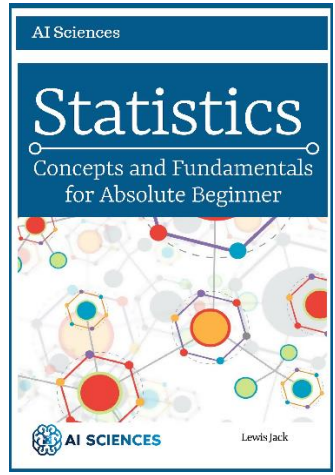
- Website: www.aisciences.net
- Email: contact@aisciences.net

From AI Sciences









WWW.AISCIENCES.NET

eBooks, free content, and online learning courses.

Did you know that AI Sciences offers free eBook versions of every book published? Please subscribe to our email list to find out about our free eBook promotion. Get in touch with us at contact@aisciences.net for more details.

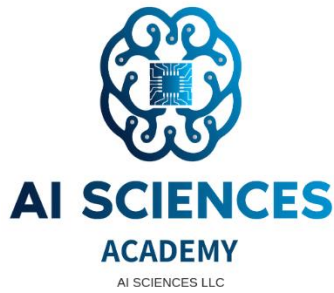
At www.aisciences.net, you can also read a collection of free books and receive exclusive free ebooks.

WWW.AISCIENCES.NET

Did you know that AI Sciences also offers online courses?

We want to help you with your career and to take control of your future with powerful and easy-to-follow courses in data science, machine learning, deep learning, statistics, and all artificial intelligence subjects.

Most courses in data science and artificial intelligence simply bombard you with dense theory. Our courses do not overload you with complex mathematics. Instead, they focus on building up your understanding of the subject for infinitely better results down the line.



Please visit our website and subscribe to our email list to be the first to know about our free courses and promotions. Get in touch with us at academy@aisciences.net for more details.

Preface

Why are AI Sciences Books Different?

AI Sciences books explore every aspect of artificial intelligence and data science using computer science programming languages such as Python and R. Our books are the best first step for beginners; they are step-by-step guides for any person who wants to start learning artificial intelligence and data science from scratch. They will help you build a solid foundation so that any subsequent high-level courses will be easy for you.

Who This Book Is For

This book is designed for students and learners who want to demystify the concepts, statistics, and math behind machine learning algorithms, and who are curious to solve real-world problems using machine learning. This book is structured to start with the basics, and then to gradually develop an understanding of the array of machine learning algorithms.

It is recommended that you have some understanding of statistics, and basics math before starting this book. However, if you are new to these subjects, it is not a limitation.

Why this Book?

This book will guide you through machine learning step by step, starting with the very basics to what machine learning is. The best part about this book is its structure; it is structured in such a way that makes the concepts easily understandable. It

will help you to understand the basics of machine learning and master them in ONE DAY! This ensures that no prior knowledge is required to start learning from this book. The content of this book is specially designed to encompass all the concepts that come under the domain of machine learning. This book not only guides you through the problems and concepts of machine learning but also elaborates how to successfully implement those concepts.

Chapter 1: Introduction to Machine Learning

What Is Machine Learning?

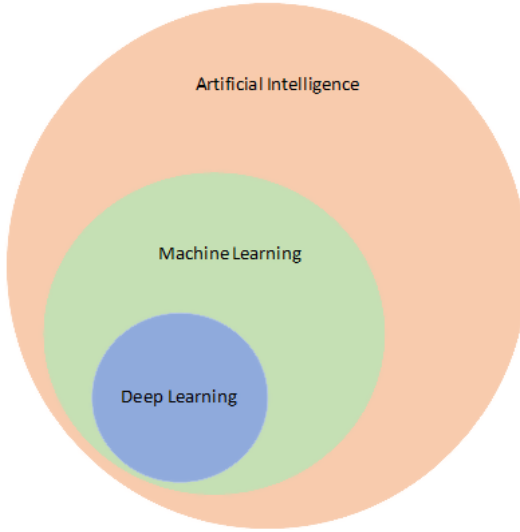
Many of the current computer systems use fixed logic to interpret input data and calculate the output. In this continuously changing world, it is difficult to maintain these systems; the accuracy of the output has also become questionable because the data for which the logic was built has drastically changed over time. Use of internet devices is increasing day by day, and these devices are generating a massive amount of structured and unstructured data every hour. Old computer systems with static rules are incapable of processing this data or extracting information from it. The availability of computation power and machine learning algorithms are paving the way for data-driven systems, which will continuously learn and evolve by themselves.

The term *machine learning* was coined by Samuel Arthur in 1959. He developed a checker-playing program that observed positions in the game and aqcuired the ability to play better moves on the part of the machine player. The program was able to improve performance over time with each game played.

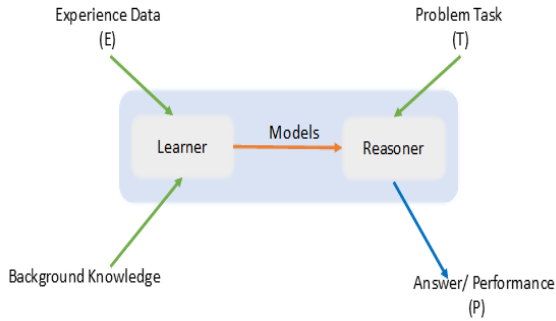
Learning is defined as the "ability to improve one's behavior with experience." machine learning, in essence, means enabling a machine/system to learn from its past experiences and improve. Machine learning algorithms can find a solution on how to complete tasks based on generalizations from historical

data and subsequently improve their performance from the experience of past data.

Machine learning is a subset of artificial intelligence (AI). It is a combination of statistical models and algorithms that enable a computer system to learn and improve from experience without being explicitly programmed.



A widely accepted machine learning definition is given by Tom Mitchell: "A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P** if its performance at tasks in **T**, as measured by **P**, improves with experience **E**."[i]



Tackling any machine learning problem consists of the following four-step process. We will discuss these in detail in further sections of the book.

- **Data preparation** – Garbage in and garbage out. Machine learning models evolve with learning from experience. If the data fed to a machine learning model is not accurate, the model will not acquire proper knowledge from the data and will inaccurately calculate output. Hence, data preparation is one of the crucial steps in building machine learning models. Finalizing an accurate dataset is an iterative process with many loops. Data preparation can be further divided into three steps.
    - **Select data** – Select the right subset of the available data. Working on all available information may not be feasible or fruitful.
    - **Preprocess data** – Selected data may not be in a suitable format to work with. In that case, it needs to be converted into a format that the machine learning model will accept. Removing noise, handling missing data, handling outliers,

etc. would help to prepare the dataset for good model learning.

    - **Transform data** – Implement the domain knowledge to get the most out of the dataset. Implement multiple transformations on preprocessed data before finally reaching a conclusion. Some common data transformation techniques are scaling, attribute decomposition (splitting features), and attribute aggregation (joining features). This step is also known as feature engineering.

- **Training set generation** – Split the dataset prepared in the previous step into one training and one test dataset. We use a training dataset to train the machine learning model, and we use a test dataset to validate the accuracy of the output. Generally, the test data accounts for ~20% of the initial dataset, but this percentage is not mandatory. It may vary as per the data at hand. It is important to have an accurate training dataset because this alone will inform the machine learning model.

- **Algorithm training** – Select an appropriate machine learning algorithm as per the dataset as well as the problem it will solve. Machine learning algorithms are divided into four major categories. We will take a deep dive into these categories in further sections of this book.

    - Supervised learning

    - Unsupervised learning

    - Semi-supervised learning
    - Reinforcement learning

- **Development and monitoring** – Once the machine learning model is developed, trained, and tested, a migration strategy needs to drafted as well as a plan for how it will evolve with time; for example, how frequently the machine learning model needs to be retrained so that it can calculate the output correctly.

Problems that Machine Learning Can Solve

Lately, machine learning algorithms have become accepted in a range of industries and consumer areas. Adaptive learning and continuous improvement have enabled machine learning to play a key role in the evolution of commercial, social, and educational domains. Machine learning has become a part of daily life. Be it getting an automatic recommendation for what videos to watch, what product to order, or what food to eat, to fingerprint and facial recognition or tagging friends in a digital photo, machine learning has become the backbone of many websites and devices. A successful machine learning model learns by generalizing input data and predicts accurate output for input data that it has never seen.

Below are examples of domains where machine learning is successfully implemented or could help in making data-driven decisions.

Medicine

- Learn from historical medical records to predict which patient will respond to which treatment.
- Disease diagnosis – Data such as symptoms, lab measurements and results in DNA tests, etc. automatically identify which kind of disease a patient has. For example, a retina scan can reveal what level of diabetes a patient has, and cancer can be identified from X-ray scans.

Vision

- Digitize handwritten scripts
- Number plate identification of a moving car
- Facial recognition for detection or unlocking a mobile device
- Identifying what an object represents in an image and where it is presented
- Self-driven cars – Analyze video streams to identify surrounding objects, their size, and speed, classify them, and take a corrective course of action to drive safely.

Autonomous Robots

- Autonomous robots that will learn to navigate from their own experience

Fraud Detection

- Credit Cards – Analyze the spending pattern of an individual, report if there is any spike, and classify it based on historical patterns as fraudulent or non-fraudulent activity.

Natural Language Processing

- Sentiment analysis – Analyze product/movie reviews, understand the context, do sentimental analysis and classify them positive or negative.
- Speech recognition – More than 1000 languages are spoken around the world. Automatic translation engines are already improving communication.
- Chatbot – Analyze customer input (text or speech), understand the context, and reply with an appropriate answer/solution.

Finance

- Stock market and share price prediction – Find out patterns and trends from historical data and analyze published news from various sources to predict how the stock market will behave.

Meteorological

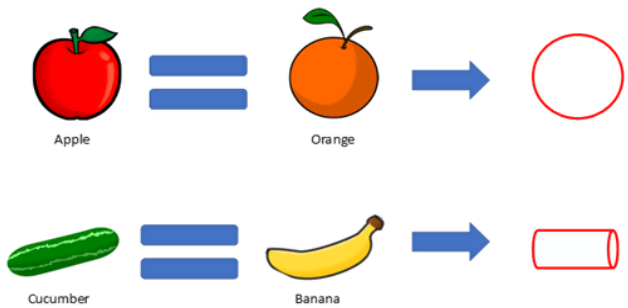
- Taking a data-driven approach to identify how the climate is changing and accurately forecast cyclones, earthquakes, hurricanes, etc. can save lives.
- Weather forecast – There are many industries that rely on certain weather conditions to be operational. An

accurate weather forecast will help them to manage resources and time efficiently.

Chapter 2: Types of Learning

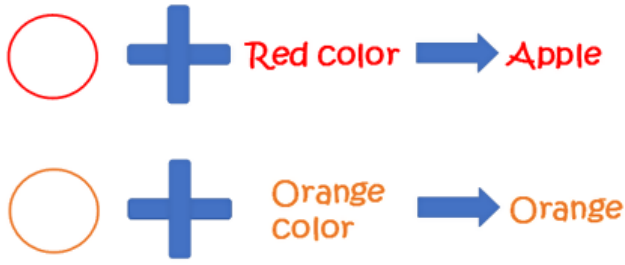
Before we jump into how a machine learning algorithm learns, let us first try to understand how a human baby learns. Think of a one-year-old human baby; the baby does not know the difference between an apple and an orange. For him, all fruit is the same, be it an orange, apple, banana, cucumber, or any other fruit for that matter. In his first phase of learning, he builds an intuition that oranges and apples are of one shape, and bananas and cucumbers are of another shape.

Phase 1 Learning



Once the baby is comfortable with the shapes of fruit, he is introduced to another property: color. Now he knows that a fruit round in shape and red in color designates an apple and a round shape and orange color belong to an orange. Similarly, he would now be able to distinguish between a banana and a cucumber.

Phase 2 Learning



Now the baby can clearly distinguish between round shaped and cylindrical shaped fruits. But to reach this stage, the baby was told numerous times that a round and red fruit is an apple, and that a round and orange fruit is an orange. Here is a subset of the information given to the baby in a tabular format:

Row#	Shape	Color	Fruit
1	Round	Orange	Orange
2	Round	Red	Apple
3	Round	Orange	Orange
4	Round	Red	Apple
5	Round	Red	Apple
6	Round	Orange	Orange
7	Round	Red	Apple

In this scenario, the fruit type is dependent on two properties: shape and color. The baby was told about the combinations of

these properties again and again until he learned how to identify the fruit type from these properties.

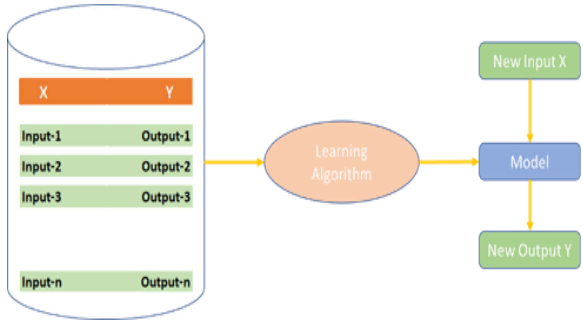
Machine learning models learn in the same way. In machine learning, the properties of the fruit, such as shape and color, are called the **features**; the fruit type is called the **label**; and each instance of an input-output pair is called an **observation**.

		Features		Label
	Row#	Shape	Color	Fruit
Observations	1	Round	Orange	Orange
	2	Round	Red	Apple
	3	Round	Orange	Orange
	4	Round	Red	Apple
	5	Round	Red	Apple
	6	Round	Orange	Orange
	7	Round	Red	Apple

Depending on the features and label passed to a machine learning algorithm, learning is classified into five categories: supervised, unsupervised, reinforcement, semi-supervised, and instance-based.

Supervised Learning

As the name suggests, this kind of learning is supervised by the trainer. Machine learning algorithms are trained with labeled observations, i.e., for each observation of training data, the input and output are known. As in our previous example, shape and color are input features, and the fruit type is the output label.



Supervised learning can also be described as "learn from the past to predict the future."

Supervised learning algorithms receive both the input features and the correct corresponding output. With each iteration, the algorithm learns by minimizing the discrepancy between correct output and predicted output. To minimize errors, the model is modified by the algorithm with each iteration.

Supervised learning algorithms primarily identify patterns from labeled data, and fit these patterns to find labels for unlabeled data.

Supervised learning is the most commonly used and successful type of machine learning so far. The downside is that it often requires human effort to label the training dataset. Supervised

learning is used in applications where past data forecast future actions.

For instance, it can predict when credit card transactions are possibly fraudulent, or which insurance customer is expected to file a claim, or how much inventory an industry should maintain to meet future customer demands.

Supervised learning is further divided into three categories.

- **Classification** – The learning dataset **label** is divided into two or more classes and the learner produces a model to assign unseen inputs into one or more of these classes. When the learning dataset label has exactly two categories, it is called binary classification, and when it has more than two categories, it is called multi-class classification.

  For example, classifying a patient as positive or negative for a disease is binary classification, whereas grading student exams with an A, B, C, or D grade is multi-class classification.

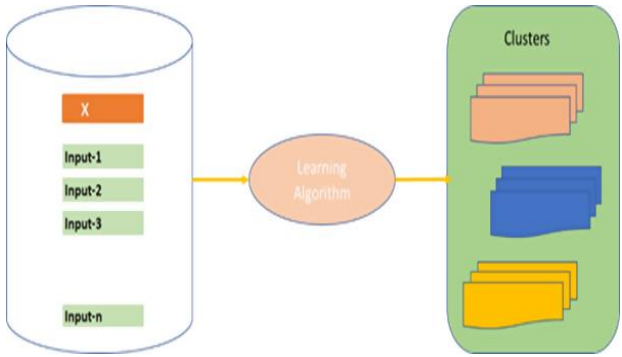
  In binary classification one class is termed as positive and another class as negative. Here, the positive class does not represent profit or benefit but rather represents the object in the study.
- **Regression** – The learning dataset label contains real numbers, and the machine learning algorithm produces a model to assign a real number to unseen inputs. In this type of learning, the output label can be any numeric value within a given range.

Creating a model to predict house prices is an example of regression. Here, the learning dataset will have multiple observations along with a real number assigned as a house price for each observation. The output of this regression model will also be a real number as a house price.

- **Anomaly detection** – Sometimes the goal is to identify the data points that are simply unusual. For example, in fraud detection, any highly unusual credit card spending pattern is considered to be suspicious. There are many probable variations compared with very few training examples. As a result, it is hard to learn what a fraudulent activity looks like. Anomaly detection takes a history of non-fraudulent transactions to determine what normal activity looks like, and then identifes anything that is notably different.

Unsupervised Learning

In this kind of learning, the trainer does not provide labeled output in the learning dataset. The machine learning algorithm learns from unlabeled data and gathers information from it.



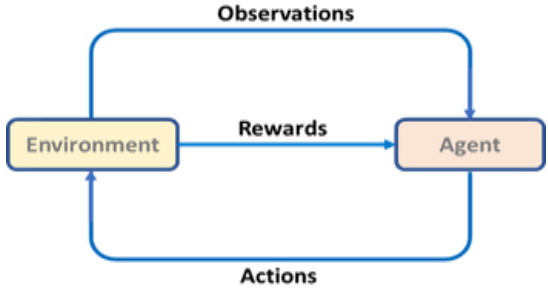
Unsupervised learning algorithms are given a learning dataset where the output is not defined for any observation present in the dataset. Unsupervised learning algorithms make clusters according to similar observations in the learning dataset. There are many successful applications of unsupervised learning models. However, these are harder to understand and evaluate. Unsupervised learning is useful for market segmentation, product recommendation, etc.

One of the commonly used applications of unsupervised learning is recommendations on e-commerce websites. On an e-commerce website, when we search for a product, the unsupervised machine learning model will recommend what other products you may like according to the product you searched for. This is possible because the model has created clusters based on other users' searches over a period.

Another typical application of unsupervised transformation is dimensionality reduction, where the high dimensional representation of data and features is translated in a new way to represent this data that summarizes essential characteristics of the data with fewer features.

Reinforcement Learning

This type of learning is often used in robotics, gaming, and navigation. With reinforcement learning, the algorithm identifies a policy (strategy) to get a maximum reward in the short and long term. The policy decides the action to be taken in a given situation.



This type of learning has three major components:

- Agent – The learner or decision maker
- Environment – Everything the agent interacts with
- Action – What the agent can do

In this type of learning, the agent performs an action in the environment. This action takes the environment to a new state and the environment gives a reward to the agent. This reward can be a negative or positive reward, a penalty, or nothing. From multiple iterations, the agent learns a policy on what action to take in any given state of the environment to not only optimize the short-term reward, but to optimize the overall utility of the agent in a given time horizon.

Reinforcement learning became popular in March 2016, when the AlphaGo program beat the world champion Lee Sedol in a game of Go. The program analyzed millions of games data and played many games against itself to learn the winning strategy.

Semi-supervised Learning

In semi-supervised learning, machine learning algorithms learn from a combination of labeled and unlabeled learning datasets. Semi-supervised learning is used for the same applications as supervised learning. It is useful when collecting a labeled dataset is costly. In this scenario, a learning dataset will have a small amount of labeled data with a large amount of unlabeled data.

Most semi-supervised learning algorithms are a combination of supervised and unsupervised learning algorithms. These algorithms create clusters from the unlabeled data and then utilized labeled data to label those clusters.

A commonly used application of semi-supervised learning is tagging friends on Facebook or Google photos. Once you upload multiple photos of the same person, the algorithm clusters them according to facial similarity, e.g., person A's photos in one cluster and person B's photos in another. Now if person A is tagged in any of the photos, all the photos in the first cluster will be labeled as "person A."

Instance-Based Learning

Up until now, the learning methods we discussed are all considered model-based learning. In model-based learning, a training dataset (learning dataset) is used to create the model and once the model is created, it does not refer to the training dataset again.

Instance-based learning never undertakes the training phase and does not create any model. The learner simply stores the training data instead of generalizing the examples and coming up with a target function.

Instance-based learning is also referred to as "lazy learning" because the processing is delayed until a new instance needs to be classified.[ii] When a new instance is encountered, its relationship to the stored training examples is examined to assign an output to the new instance. This processing delay leads to more time consumption during the prediction phase.

The key advantage of instance-based learning is that it is more dynamic than model-based learning. Model-based learning methods create a generic target function for the entire space, but instance-based learning methods can estimate the target function differently for each new instance to be classified.

Chapter 3: Data Structures and Linear Algebra

Machine learning is about creating mathematical formulas to predict the future based on historical events. In this chapter, we will discuss the math that is required to understand machine learning algorithms. This understanding will help us to perform model evaluations and to comprehend why one model may behave better than another in a given scenario.

Machine learning is powered by linear algebra, calculus, and probability and statistics. Let us discuss them one by one.

Notation

Machine learning involves a lot of mathematical calculations in each step of learning and prediction. Be it dataset preparation or algorithm optimization, many mathematical transformations and equations are executed to reach to a final solution. The most commonly used machine learning notations that we will cover in this book are listed in the following table.

	Symbol	Name	Description	Example
Algebra	(f∘g)	composite function	a nested function	(f ∘ g)(x) = f(g(x))
	Δ	delta	change / difference	Δx = x_1 - x_0
	e	Euler's number	e = 2.718281828	s = frac{1}{1+e^{-z}}
	∑	summation	sum of all values	∑ x_i = x_1 + x_2 + x_3
	∏	capital pi	product of all values	∏ x_i = x_1·x_2·x_3
	ϵ	epsilon	tiny number near 0	lr = 1e-4
Calculus	x'	derivative	first derivative	(x^2)' = 2x
	x''	second derivative	second derivative	(x^2)'' = 2
	lim	limit	function value as x approaches 0	
	∇	nabla	gradient	∇f(a,b,c)
Linear algebra	[]	brackets	matrix or vector	M=[135]
	•	dot	dot product	(Z=X•W)
	⊙	hadamard	hadamard product	A=B⊙C
	X^T	transpose	matrix transpose	X^T•W
	$\vec{x}$	vector	vector	v=[123]
	X	matrix	capitalized variables are matrices	X,W,B
	$\hat{x}$	unit vector	vector of magnitude 1	x^=[0.20.50.3]
Probability	P(A)	probability	probability of event A	P(x=1) = 0.5
Set theory	{ }	set	list of distinct elements	S = {1, 5, 7, 9}
Statistics	μ	population mean	mean of population values	
	$\overline{x}$	sample mean	mean of subset of population	
	σ^2	population variance	variance of population value	
	s^2	sample variance	variance of subset of population	
	σ_X	standard deviation	population standard deviation	
	s	sample std dev	standard deviation of sample	
	ρX	correlation	correlation of variables X and Y	
	$\tilde{x}$	median	median value of variable x	

Data Structure

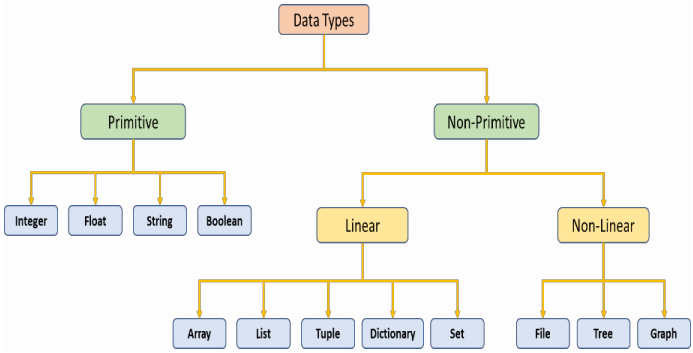
So far, we have discussed in detail the role that data plays in machine learning. But what structure is that data presented in?

Data structure is an organized form of representing and storing data, which helps when accessing and maintaining data. Data structures are designed to organize data to fulfill a specific

purpose, so that the data can be accessed and worked with in the appropriate way.

Data structures are mainly grouped in two types:

1. **Primitive data types** – These are the core data types that a coding language system understands.
2. **Non-Primitive data types** – These data types are *derived* from primitive data types.



	Data Type	Properties	Example
Primitive	Integer	An integer number	1, 2, 3
	Float	A real number	1.2, 4.9
	Boolean	True or False condition	True, False
	String	• Sequence of characters • Immutable	"My name is Alex"
	Array	A list of elements of same data types Can have one or more dimensions	$\begin{bmatrix} 2, 5, 7, 9 \\ 3, 6, 8, 4 \end{bmatrix}$

non-Primitive		Mutable	**[**car, bike, truck, auto**]** **[**23, 43, 95, 48**]**
	List	• A list of elements • Mutable • Can have elements of multiple data types in one list • Index is maintained	**[**cat, dog, mouse**]** **[**cat, 25, 36, True**]**
	Tuple	• A list of elements • Immutable • Can have elements of multiple data types in one list • Index is maintained	**(**car, bike, truck, auto**)** **(**car, 4, 90, bhp**)**
	Dictionary	• Set of key value pairs • Mutable • Values are accessed via keys • Index is not maintained	**{** type: car brand: Honda bhp: 120 **}**
	Set	• Unordered collection of unique elements • Cannot have duplicate values • Can have elements of multiple data types in one set • Mutable • Index is not maintained	**{**car, bike, auto**}** **{**1, 5, 2, 8**}**
	File	Unstructured data type to store data	csv, tsv, excel
	Tree	• Each tree has one root node	

		• Each node except root node is associated with one parent node • Parent node can have multiple child nodes	
	Graph	Pictorial representation of set of objects	

A data type is said to be immutable when values of variables of this data type cannot be changed, e.g., values of variables t1 of the data type Tuple cannot be changed:

t1 = (1, 2, 4, 5)

Sets, Vectors, and Matrices

Sets and vectors are building blocks for statistics and machine learning algorithms. To understand the math behind an algorithm, we will look at sets, vectors, and matrices.

Set: A set is defined as an unordered collection of unique elements. Sets can have heterogeneous elements, but one element cannot be repeated in the same set. Sets are expressed as **auto = {1, 4, 6, 9, a, c}.** Here, *auto* is the name of the data type variable, and it has six elements: four numbers (1, 4, 6, and 9) and two letters (a and c).

Set Operations

Consider the following two sets in order to understand the set operations:

s1 = {1, 4, 6, 8, 12, 18, 23}

s2 = {9, 5, 3, 1, 4, 19, 43}

Union (∪) – The union of two given sets is the smallest possible set that contains all the elements of both the sets. For example, the union of s1 and s2 will consist of all the elements of s1 and all the elements of s2, but without repeating any element. The symbol for denoting the union of sets is **∪**.

s1 ∪ s2 = {1, 3, 4, 5, 6, 8, 9, 12, 18, 19, 23, 43}

Intersection (∩) – The intersection of two given sets is the set that contains all the elements that are common to both sets. The symbol for denoting the intersection of sets is ∩.

s1 ∩ s2 = {1, 4}

Difference – The difference of s1 and s2 is a set of elements that are present in s1 but not in s2, and vice versa. It is denoted by the symbol −.

s1 − s2 = {6, 8, 12, 18, 23}

s2 − s1 = {3, 5, 9, 19, 43}

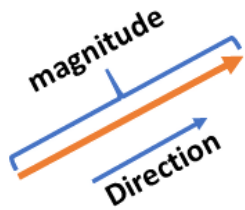
Symmetric Difference – The symmetric difference of s1 and s2 is a set of elements present in both s1 and s2, except those that are common in both. It is represented by the symbol Δ.

s1 Δ s2 = {3, 5, 6, 8, 9, 12, 18, 19, 23, 43}

Complement – The complement of set s1 is a set of all elements except those that are present in set s1.

Vectors

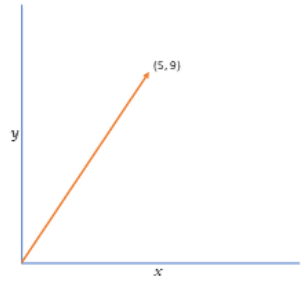
In physics, a scalar is defined as something that has value (magnitude) but no direction attached to it. A vector is defined as something that has both magnitude and direction, where magnitude is the length of the vector and orientation is the direction of the vector (e.g., speed is a scalar since it does not have an implied direction, but velocity is a vector since it has both magnitude and direction).



In machine learning, a vector of dimension n is defined as an ordered collection of n elements. In Python vectors are created as an array (an ordered homogeneous sequence of elements).

Each vector represents a point in a number space. This point could be in a two-dimensional, three-dimensional, or n-dimensional space, e.g.:

a = [5, 9]



Each element in a vector has an associated index. In the above example, the vector element 5 can be accessed as a[0] and the element 9 can be accessed as a[1].

Matrix

Two-dimensional arrays are called matrices. If a matrix has m rows and n columns, the order of the matrix is **m x n**.

$$X = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ 6 & 5 & 9 \end{bmatrix}$$

Each element in a matrix has an associated index. In the above example, the element -8 can be accessed as $X[1][2]$.

If a matrix has the same number of rows and columns, then it is called a square matrix (the above matrix X is a square matrix, for example).

Vector and Matrix Operations

Let us consider the following vectors and matrices in order to understand vector and matrix operations:

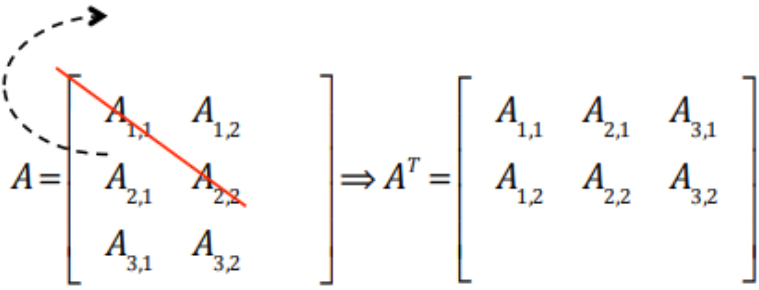
v1 = [6, 8, 13, 45, 32, 6]

v2 = [9, 4, 12, 4, 27, 32]

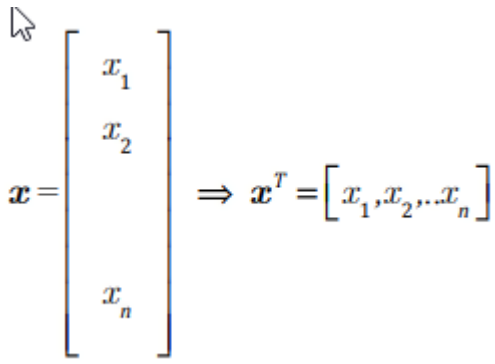
$$A=\begin{bmatrix}1 & 2\\ 5 & 6\end{bmatrix} \qquad B=\begin{bmatrix}3 & 7\\ 9 & 2\end{bmatrix} \qquad C=\begin{bmatrix}4 & 7 & 2\\ 5 & 6 & -1\end{bmatrix}$$

Transpose – The transpose of a matrix is defined as a mirror image across a diagonal line. The transpose of matrix A is denoted as A^T:

$$A=\begin{bmatrix}A_{1,1} & A_{1,2} & A_{1,3}\\ A_{2,1} & A_{2,2} & A_{2,3}\\ A_{3,1} & A_{3,2} & A_{3,3}\end{bmatrix} \Rightarrow A^T=\begin{bmatrix}A_{1,1} & A_{2,1} & A_{3,1}\\ A_{1,2} & A_{2,2} & A_{3,2}\\ A_{1,3} & A_{2,3} & A_{3,3}\end{bmatrix}$$



$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} A_{1,1} & A_{2,1} & A_{3,1} \\ A_{1,2} & A_{2,2} & A_{3,2} \end{bmatrix}$$



$$x = \begin{bmatrix} x_1 \\ x_2 \\ \\ \\ x_n \end{bmatrix} \Rightarrow x^T = \begin{bmatrix} x_1, x_2, .. x_n \end{bmatrix}$$

Addition – Vector addition is possible only when both the vectors are the same size.

v1 + v2 = [15, 12, 25, 49, 59, 38]

v1 + 3 = [9, 11, 16, 48, 35, 9]

Matrix addition is possible only when both matrices are the same size.

A + B = $\begin{bmatrix} 4 & 9 \\ 14 & 8 \end{bmatrix}$

Subtraction – Vector subtraction is possible only when both vectors are the same size.

v1 - v2 = [-3, 4, 1, 41, 5, -26]

v1 - 3 = [3, 5, 10, 42, 29, 3]

Matrix subtraction is possible only when both matrices are the same size.

$$A - B = \begin{bmatrix} -2 & -5 \\ -4 & 4 \end{bmatrix}$$

Comparison – The comparison of two vectors shows how different two vectors are.

v1 = v2 = [False, False, False, False, False, False]

v1 > v2 = [False, True, True, True, True, False]

Multiplication: Vector multiplication is supported only for vectors of the same size.

Scalar multiplication

v1 * 3 = [18, 24, 39,135, 96, 18, 45, 87]

Vector multiplication

v1 * v2 = [54, 32, 156, 180, 864, 192, 345, 522]

Dot product: This can be done between two vectors of equal length. It will result in a scalar value.

a = [a1, a2, a3, a4] b = [b1, b2, b3, b4]

$a.b = \sum a_i * b_i$

v1.v2 = 2345

Matrix multiplication: Matrix multiplication is possible only when the number of columns in the first matrix is equal to the number of rows in the second matrix. If matrix A is of order (m, n) and matrix B is of order (r, s), A X B is possible if n=r. The resulting matrix will be of order (m, s).

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} X \begin{bmatrix} u & v \\ w & x \\ y & z \end{bmatrix} = \begin{bmatrix} au+bw+cy & av+bx+cz \\ du+ew+fy & dv+ex+fz \end{bmatrix}$$

$$AXB = \begin{bmatrix} 21 & 11 \\ 69 & 47 \end{bmatrix}$$

Matrix multiplication properties

Associativity: A(BC) = (AB)C

Distributivity: A(B + C) = AB + AC

AB ≠ BA

The dot product between vectors is cumulative: $\mathbf{x}^T\mathbf{y} = \mathbf{y}^T\mathbf{x}$

(AB)T = ATBT

Identity matrix: An identity or unit matrix of size n is the square matrix of order n where all the diagonal elements are 1s and all the other elements are 0s. It is denoted by ***I.***

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The multiplication of a matrix with its inverse results in an identity matrix. When a vector is multiplied with an identity matrix, it does not change its value.

$$A^{-1}A = I_n$$

Functions

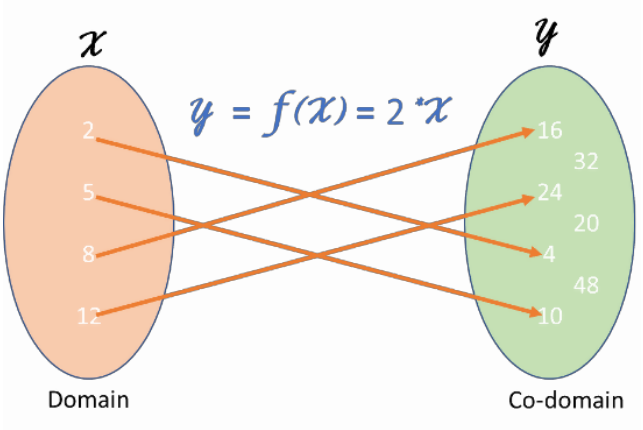
Function is the relation between an input and an output, where each input is related to only one output. The set of allowable input values is called the domain and the set of all possible output is called the co-domain of the function. The set of output values corresponding to each domain value is called the range of the function.

In the following example,

domain = [2, 5, 8, 12];

co-domain = [16, 32, 24, 20, 4, 48, 10]; and

range = [16, 24, 4, 10].

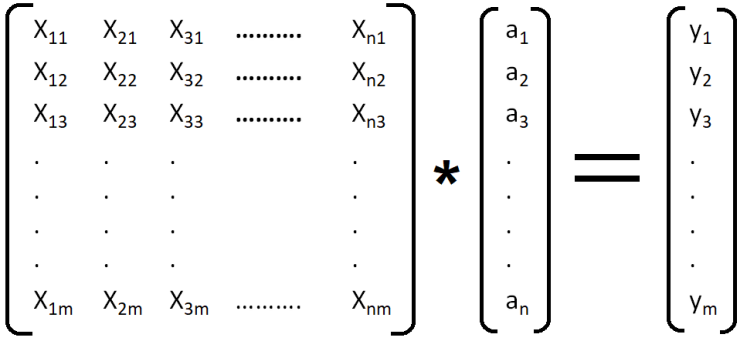


In machine learning, all of the algorithms use matrix multiplication for function calculation.

For instance, a function is defined as:

$y = a + a_1X_1 + a_2X_2 + a_3X_3 + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + a_nX_n$

Here, a, a_1, a_2, …, a_n are constants and X_1, X_2, X_3…. X_n are variables. For each record in a set of m records, function can be represented as:



Derivative and Gradient

Rate of change: Rate of change is a measure of how a variable is changing with respect to another variable. For example, if you are traveling to your home by car, speed is the measure of how the distance between your current position and home is changing with respect to time.

Rate of change is measured in two ways:

Average rate of change: This measure represents the average rate of change for a function, e.g., if the distance between your home and office is 40 kilometers and it takes you 40 minutes to reach home, then the average rate of change in distance (speed) is:

40 km/40 min = 1 km/min

Instantaneous rate of change: This measure states the rate of change at a position or moment, e.g., on your way home, if the relation between distance and time is represented as a function of time (**f(t)**), then the derivative of the function (**f(t)**) with respect to

time (**t**) is the rate of change at a moment (car speed at that moment).

In math, this is denoted as:

$$f'(t) = \lim_{\Delta t \to 0} \frac{\Delta f(t)}{\Delta t}$$

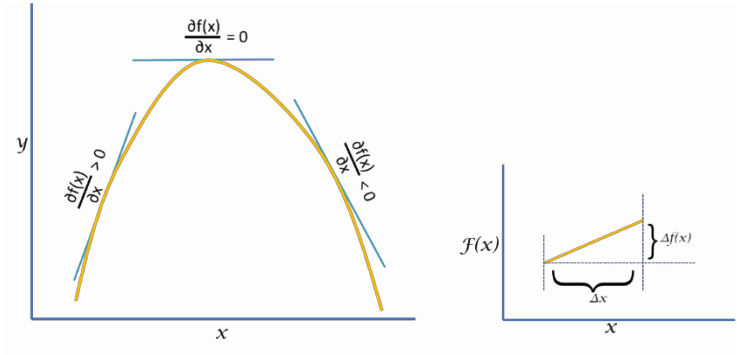
Δt is a very small change in time and is approximately equal to zero.

Derivative: The derivative of a function (y = f(x)) with respect to x represents how much the dependent variable (**y**) changes for a delta change in domain value (**x**). In other words, it is known as the rate of change of a function.

It is denoted as:

$$f'(x) = \frac{\partial f(x)}{\partial x}$$

As shown in the below figure, Δf(x) is the change in function f(x) for Δx change in x. It is also represented as a slope at that value of variable x. If Δf(x) is positive, then it means the slope is inclining and the function curve is rising. If Δf(x) is negative, then it means slope is declining and the function curve is falling. If Δf(x) is zero, then it means the slope is not changing and the function curve is constant.



Derivatives of commonly used function types are presented in the following table.

1. $\frac{d}{dx}(x) = 1$
2. $\frac{d}{dx}(ax) = a$
3. $\frac{d}{dx}(x^n) = nx^{n-1}$
4. $\frac{d}{dx}(\cos x) = -\sin x$
5. $\frac{d}{dx}(\sin x) = \cos x$
6. $\frac{d}{dx}(\tan x) = \sec^2 x$
7. $\frac{d}{dx}(\cot x) = -\csc^2 x$
8. $\frac{d}{dx}(\sec x) = \sec x \tan x$
9. $\frac{d}{dx}(\csc x) = -\csc x(\cot x)$
10. $\frac{d}{dx}(\ln x) = \frac{1}{x}$
11. $\frac{d}{dx}(e^x) = e^x$
12. $\frac{d}{dx}(a^x) = (\ln a)a^x$
13. $\frac{d}{dx}(\sin^{-1} x) = \frac{1}{\sqrt{1-x^2}}$
14. $\frac{d}{dx}(\tan^{-1} x) = \frac{1}{1+x^2}$
15. $\frac{d}{dx}(\sec^{-1} x) = \frac{1}{|x|\sqrt{x^2-1}}$

1. $(cf)' = cf'(x)$
2. $(f \pm g)' = f'(x) \pm g'(x)$
3. $(fg)' = f'g + fg'$ – **Product Rule**
4. $\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$ – **Quotient Rule**
5. $\frac{d}{dx}(c) = 0$
6. $\frac{d}{dx}(x^n) = nx^{n-1}$ – **Power Rule**
7. $\frac{d}{dx}(f(g(x))) = f'(g(x))g'(x)$
   This is the **Chain Rule**

Gradient – Gradient gives the rate of change of a function in every direction (= number of variables). It has both magnitude and direction; hence, is presented as a vector. Gradient helps when calculating the slope at a specific point on a curve for functions with multiple independent variables. It points in the direction of the greatest rate of increase of the function, and its magnitude is the slope of the curve in that direction.

Gradient stores the partial derivatives of multivariable functions. To calculate this slope, we need to isolate each

variable to determine how it impacts the output on its own. To do this, we iterate through each of the variables and calculate the derivative of the function after holding all other variables constant. Each iteration produces a partial derivative, which we store in the gradient.

Gradient is represented as:

$$\nabla f(x,y) = \left(\frac{\partial f(x, y)}{\partial x} , \frac{\partial f(x, y)}{\partial y} \right)$$

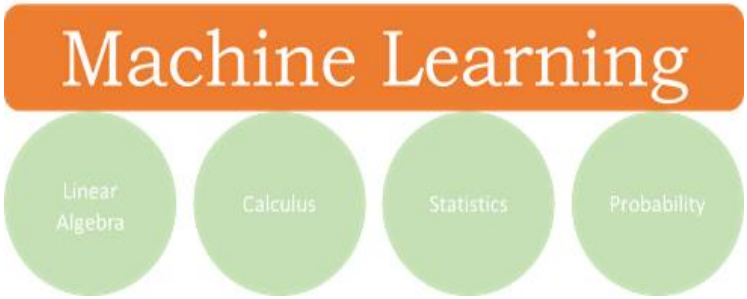
e.g., for a multivariate function:

$f(x, y) = 2x^2y^3 + x^3 + y^3$

$\nabla f(x, y) = (4xy^3 + 3x^2 , 6x^2y^2, 3y^2)$

Chapter 4: Statistics and Probabilities

Machine learning is powered by linear algebra, calculus, and statistics and probability. In this chapter, we will discuss statistics and probability and what role they play in the machine learning domain.



What is Statistics?

Statistics is the science of collecting, organizing, presenting, analyzing and interpreting data to help in making more effective decisions.[iii] We will start by understanding how statistics plays a major role in machine learning.

Statistical analysis is applied to manipulate, summarize, and investigate data so that useful decision-making information results are obtained.

Suppose you are running an online cloud storage company. You want to send a campaign to customers, but you have questions: to whom you should send the campaign? How many customers should you send the campaign to? How effective

will the campaign be? Will the person who receives the message buy more than the person who does not? Statistics is useful for getting the answers to these questions.

Statistics is classified into two types.

Descriptive Statistics

This is a method of organizing, summarizing, and presenting data in an informative way. Descriptive statistics helps us understand the data and get insights from it. For example, you want to know how many customers are coming to a store, how many of them are female or male, how many of the male customers are smokers, how many of the female customers are non-smokers, and how many of them are married. Descriptive statistics can answer all these questions.

Inferential Statistics

Usually, the collection of an entire dataset (population in statistics) is impossible. Hence, a subset of the population (also called sample) is collected, and a conclusion about the entire population is drawn. Conclusions about the population dataset are inferred from conclusions about the sample dataset. For example, let us imagine that elections are upcoming in a particular state, and we want to know which party is going to win. We cannot ask each person in the state their opinion on which party will win. Instead, we ask a few people—a sample group—who can represent all segments of the population of the state. Based on their answers, we will deduce which party is going to win the election. The population, in this case, is the entire population of the state, and the sample is the set of people who were polled.

Before jumping into the ocean of statistics, we will start by defining some generic terms used in statistics.

Introduction to Basic Terms

Here are a few terms that are used quite often in statistics.

Population

A population is the entire set of data for which we want to perform statistical analysis. For instance, if we want to study stars, then all the stars in the universe make up the population.

Population is further categorized into two classes:

1. Infinite population – A dataset where all data points cannot be counted, such as the stars in the universe.
2. Finite population – A dataset where all data points can be counted, such as all the subscribers of a telecom company.

Sample

A sample is a subset of the population. If we select a few stars from the population (universe) to study, then that set of selected stars is called a sample.

	Population	Sample
Size	N	n
Mean	μ	$\bar{x}$
Variance	σ^2	s^2
Standard Deviation	σ	s

Variable

A variable is any characteristics, number, or quantity that can be measured or counted. For example, if we want to run a campaign for customers, we want to know how many of the customers are adults, how many of them have kids, how many of them are married, what is their background, etc. All these are variables help us to understand and analyze the data. There are two types of variables:

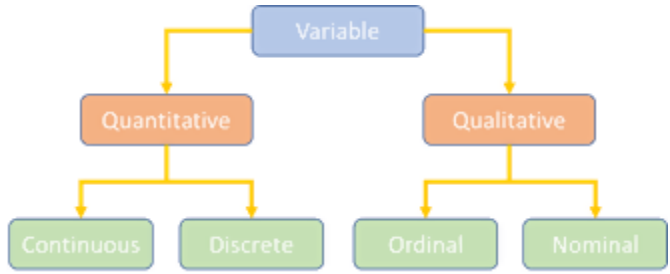
1. Quantitative or numerical variable – A variable that quantifies a population element, such as the average amount withdrawn per transaction from an ATM, the average age of students in a university, etc. Quantitative variables are further divided into two subcategories.

- Discrete – This is a whole integer number; for example, the number of people who opened a marketing campaign email.
- Continuous – This can be any numerical value; for example, the average age of an employee in an enterprise.

2. Qualitative or categorical variable – A variable that categorizes or describes a population element, e.g., red, yellow,

or green; male or female, etc. Qualitative variables are further categorized into:

- Ordinal – Discrete values that can be compared; for example, the ratings given to a cab driver by two passengers.
- Nominal – Discrete values that cannot be compared; for example, male or female, yellow or blue, etc.



Data

Data refers to values collected for the variable from each of the elements belonging to the sample or population. For example, a person carries a phone of brand X, five customers purchase $50 gift cards, etc.

Experiment

An experiment is a planned activity with results that yield a dataset, e.g., whether TV, newspaper, or social media advertisements are most effective at generating sales.

Parameter

A parameter is a numerical value that summarizes the entire population data. It is a configuration internal to the model, and its value can be estimated from data, e.g., what is the average salary of a male in the United States, what is the average age of people who own a house, etc. μ (mean) and σ (standard deviation) are parameters of a distribution.[iv]

Hyperparameter

Hyperparameter is a configuration external to the model. Its value cannot be estimated from data. There is no specific rule of thumb to identify the hyperparameter for a given problem.

Statistics

Statistics refers to a numerical value that summarizes the sample data. For example, to calculate the average salary of all males in the United States, we record the average salary in each state (sample). The average salary of each sample is a statistic.

Measures of Central Tendency

Mean, median, or mode – These are the parameters that attempt to describe a dataset by identifying the central position within the dataset. As such, measures of central tendency are sometimes called measures of central location. Measures of central tendency help to compare two datasets.

- Mean – the average of all data points in the dataset. It is represented as $\bar{x}$, and defined as:

$$\text{Mean} = \frac{\text{Sum of all data points}}{\text{No of data points}}$$

For example, in a classroom, there are 10 students and their ages are 18, 21, 19, 20, 18, 21, 22, 19, 18, and 20. The mean age of the class is:

$$\frac{18 + 21 + 19 + 20 + 18 + 21 + 22 + 19 + 18 + 20}{10} = 19.6$$

- Median – the middle value of the dataset when the data points of the dataset are arranged in ascending order.
    - If the number of data points is odd, then the median is exactly the middle value.
    - If the number of data points is even, then the median is the midway between the two middle values.

  In the above example of classroom age, the median is calculated as follows:

Total number of records is ten (even).

Sorted values: 18, 18, 18, 19, **19, 20**, 20, 21, 21, 22.

$$\text{Median} = \frac{19 + 20}{2} = 19.5$$

- Mode – the most frequently occurring value in the dataset. In the classroom age example, the modal age of the class is 18.

Important Facts about Mean, Median, and Mode

- If the dataset contains large values, then these values tend to inflate the mean. However, the median is not

significantly influenced by large values, so it is a better measure of certainty.

- If mean = median = mode, then the data is symmetrically distributed.

Measures of Dispersion

The measure of central tendency does not provide information about how the data is distributed. For example, the mean sales of two stores is the same, but the mean does not tell us which age group is buying more from the first store and which age group is buying more from the second store.

Measures of dispersion help us achieve these insights from the dataset. Measures of dispersion characterize how the data is distributed. Commonly used dispersion measures are:

- **Range** – the difference between extreme values of a dataset (max. value – min. value).
- **Variance** – a statistic that measures the closeness between data points in a dataset. It is the arithmetic mean of squared deviations from the sample mean. Variance can be calculated using the following formula:

Sample Variance

$$s^2 = \frac{\Sigma (x - \bar{x})^2}{n - 1}$$

Population Variance

$$\sigma^2 = \frac{\Sigma (x - \mu)^2}{N}$$

- **Standard deviation** – a statistic that measures the dispersion of the data around the mean. It is the square root of the variance.

ChebbySheff's theorem – a rule that provides a more general interpretation of standard deviation. It is applicable to all distributions except bell-shaped distributions. It states that the proportion of observations in any sample that lies within the *k standard* deviations of the mean is at least:

$$1 - \frac{1}{k^2} \quad for \quad k > 1$$

If k = 2, the theorem states that ¾ of all observations lie within two standard deviations of the mean.

Probability – a measure of the likelihood that an event will occur. It represents the strength of a belief. In other terms, it is a numerical way of describing how likely something is to occur.

Rules of Thumb of Probability

- The probability of an event varies between 0 and 1.
- The probability of an event that is certain to occur is 1.
- The probability of an event that is not certain to occur is 0.
- The sum of probabilities of all mutually exclusive events is equal to 1.

Tossing a coin has two outcomes: heads or tails. Since heads and tails are mutually exclusive events, they cannot occur

simultaneously. The probability of landing on heads is expressed as follows:

$$\frac{\text{outcome head}}{\text{total no. of possible outcomes}} = \frac{1}{1+1} = 0.5$$

The probability of landing on tails is expressed in the exact same way.

If each event in the sample space is equally likely, then the probability of event A occurring is:

$$\text{P (A)} = \frac{\text{(no of elements in A)}}{\text{(no of elements in sample space)}}$$

Probability is important in analyzing historical data to find a pattern, under the assumption that the past reflects the future.

Probability Rules

- **Addition**

$$\text{P (A} \cup \text{B)} = \text{P(A)} + \text{P(B)} - \text{P(A} \cap \text{B)}$$

- **Independence** – When two events are independent of each other and can occur simultaneously then the probability of both events occurring simultaneously is defined as:

$$\text{P(A} \cap \text{B)} = \text{P(A)P(B)}$$

For example, landing a coin toss on heads and the chance of rain in California are two independent

events. The probability of landing on heads (A) and rain in California (B) will be:

$$P(A) \times P(B)$$

- **Conditional Probability:** The probability of an event occurring in relation to the probability of the occurrence of a preceding event, e.g., the probability of you arriving at the office late given the probability of your train running late.

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

P(A | B) is the probability of event A occurring given that event B has already occurred.

$$P(A|B) = \frac{P(A \cap B)}{P(A \cap B) + P(A' \cap B)}$$

For example, when a red card is pulled from a standard deck of cards, what is the probability that it is also a card of hearts?

$$P(H|R) = \frac{P(H \cap R)}{P(R)}$$

The probability of pulling a red card is expressed as follows:
P(R) = 26/52 = ½ = 0.5
The probability of pulling a red card that is *also* a suit of hearts is expressed as follows:
13/52 = ¼ = 0.25

The probability of that the card is a suit of hearts given the card pulled is red:
color card = 0.25/0.5 = ½ = **0.5**

Bayes' theorem

This theorem relates the conditional and unconditional probabilities of events A and B, where B has a non-zero probability. It allows us to use one conditional probability to compute another conditional probability. It is helpful in finding out the probability of an event given that another event has already occurred.

For example, while walking in a garden, we notice that the grass is wet, and we want to find out the probability that this is a result of rain rather than a sprinkler.

Reverend Thomas Bayes derived the formula below to calculate conditional probabilities.

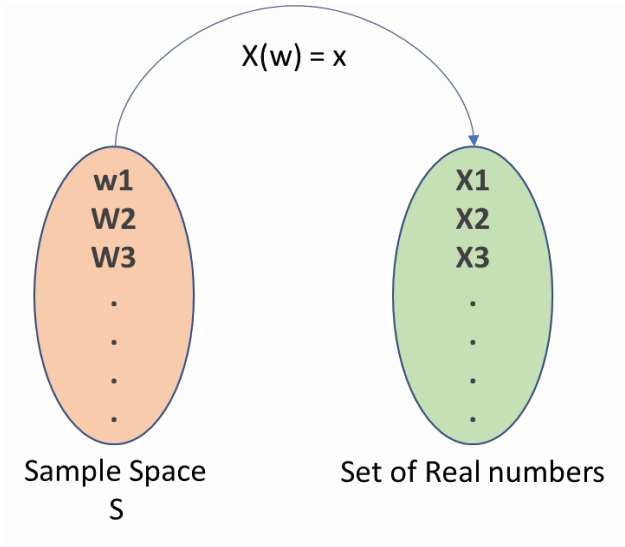
$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Here:

- A and B are events and P(B) ≠ 0
- P(A) and P(B) are probabilities of A and B occurring independently.
- P(A|B) is the probability of A occurring given that B has occurred.

- P(B | A) is the probability of B occurring given that A has occurred.
- **Random variable:** Random variable is a function or rule that maps each event in a sample space to real numbers. A random variable is denoted by **X.** If w_i is an element of sample space S and mapped to a real number X_i, then:

$$\mathbf{X}(w_i) = X_i$$

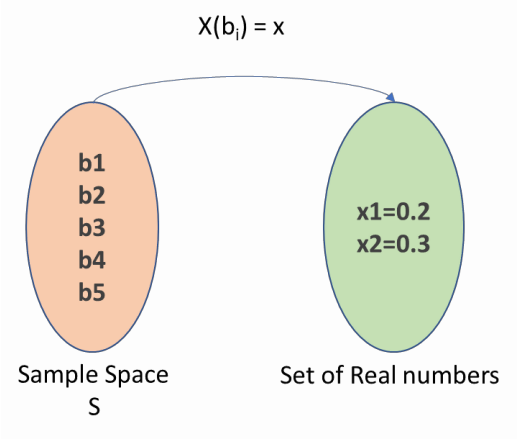


For example, suppose there are five balls—labeled b1, b2, b3, b4, and b5—in a bag. The random variable X is the weight in kg of a ball selected at random. Balls b1 and b2 each weigh 0.2 kg and balls b3, b4, and b5 each weigh 0.3 kg. This information can be represented as:

X(b1) = 0.2 X(b2) = 0.2

X(b3) = 0.3 X(b4) = 0.3

X(b5) = 0.3



$$P(X = x1) = 2/5, P(X = x2) = 3/5$$

The probability distribution of a random variable X tells us what values it can take and what the probability is corresponding to each variable.

The average of the random variable is called the expected value.

Random variables are classified into two types:

1. Discrete random variable – A whole number variable. For example, rolling a dice will produce one value in the range of 1 to 6. Probability distribution function in the case of a discrete random variable is called a probability mass function.

For example, two dice are rolled simultaneously, and the sum of both dice is a random variable. Possible outcomes are denoted as values (X_i) and the number of possible values is

denoted as frequency (n_i), and the probability of a value (X_i) occurring is denoted as probability (P_i). So:

when the total number of possible outcomes is 36 (6 x 6), and the value 2 can only occur in one way (when both dice face 1), then the probability of the outcome 2 is 1/36.

The probability distribution function for this scenario can be written as:

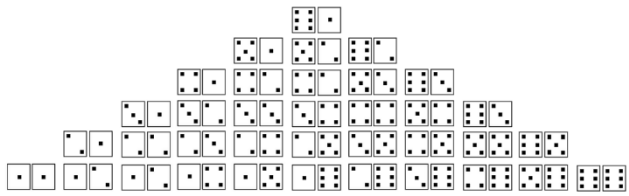
P(x = 2) = 1/36 P(x = 5) = 4/36

P(x = 3) = 2/36 P(x = 8) = 5/36

The expected value of a discrete random variable is:

$$E(x) = \sum_{i=1}^{11} p_i x_i$$

E(x) = 252/36 = 7



Value (x_i)	2	3	4	5	6	7	8	9	10	11	12
Freq (n_i)	1	2	3	4	5	6	5	4	3	2	1
Prob (p_i)	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$

Cumulative distribution function (CDF) – This is the cumulative probability of one or more events occurring, e.g.,

the probability of achieving a value lower than 5 ($x < 5$) from rolling two dice is:

$P(x < 5) = P(x = 1) + P(x = 2) + P(x = 3) + P(x = 4)$
$= 1/36 + 2/36 + 3/36 + 4/36$
$= 10/36 = 5/18$

2. Continuous Random variable – A variable that holds any real number, e.g., the time taken to travel from point A to point B. This traveling time could be 20 minutes, 2.5 hours, or any other real number for that matter.

Probability distribution function in the case of a continuous random variable is called the probability density function. The probability is calculated as the area under the probability density function.

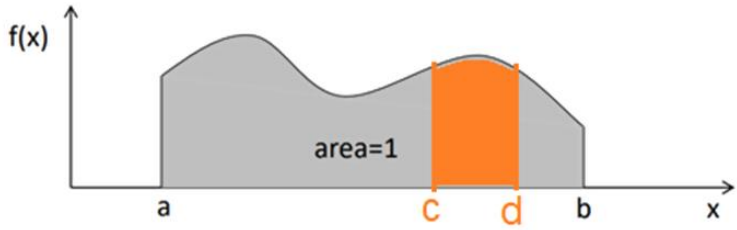
Consider an experiment where you are observing the time interval between two trains at a metro station. This time interval can be any real number, i.e., 5 minutes, 5.01 minutes, 5.0123 minutes, etc. It is difficult to tell exactly how much of an interval there will be between two trains; hence, the continuous random variable probability is calculated as the area under the curve between two points. In the above example, it is possible to calculate the probability of a train arriving between 5 and 5.0123 minutes.

The probability of random variables is calculated from the area under the curve of probability density function (PDF). For an individual value, probability is always zero since the area under a curve for a point is zero.

$$F_X(x) = \int_{-\infty}^{x} f_X(t)\, dt$$

A function f(x) is called a **probability density function** over the range $a \leq x \leq b$ if:

- $f(x) > 0$ for all values between a and b
- total area under the curve between a and b is 1



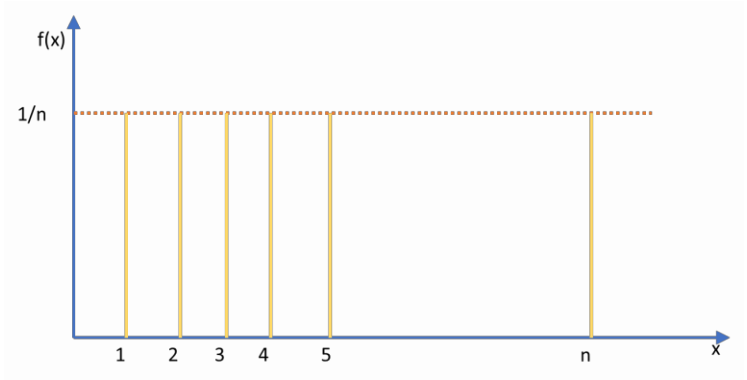
- probability between c and d is the area under the curve between c and d (shaded region)
- P(c) = 0 and P(d) = 0.

Discrete Probability Distributions

Uniform

A distribution is uniform when the probability of each outcome is the same, and all outcomes are equally likely, e.g., in the tossing of an unbiased coin, the probability of getting head or tail is 0.5.

If the number of possible outcomes (k) is known, all other parameters, i.e., mean and standard deviation, can be calculated. The number of possible outcomes (k) is called the parameter of the uniform distribution.



Sample space $S = \{1, 2, 3, \ldots, k\}$.

Random variable X defined by $X(i) = i$, $(i = 1, 2, 3, \ldots, k)$.

Distribution: $P(X = x) = \frac{1}{k}$ $\quad (x = 1, 2, 3, \ldots, k)$

$$\mu = E[X] = \frac{(1 + 2 + \cdots + k)}{k} = \frac{\frac{1}{2}k(k+1)}{k} = \frac{k+1}{2}$$

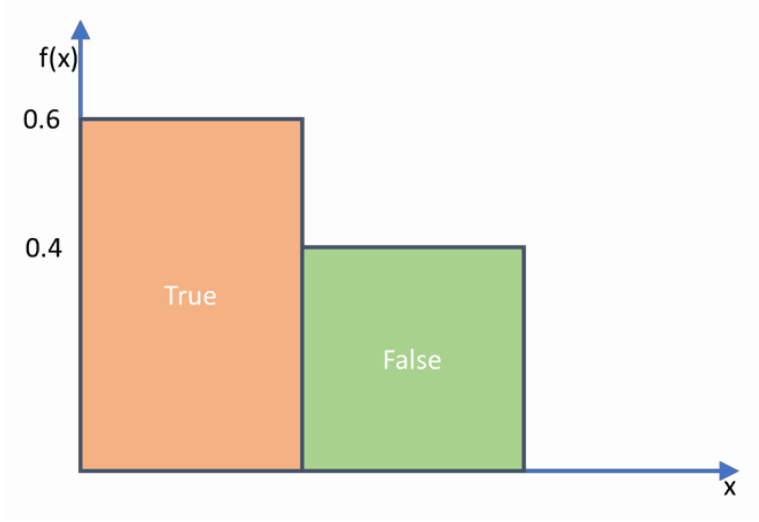
$$E[X^2] = \frac{(1^2 + 2^2 + \cdots + k^2)}{k} = \frac{\frac{1}{6}k(k+1)(2k+1)}{k} = \frac{(k+1)(2k+1)}{6}$$

$$\Rightarrow \sigma^2 = \frac{k^2 - 1}{12}$$

Bernoulli

When the outcome of an experiment has only two possible values, i.e., true and false, then the distribution is called the Bernoulli distribution, e.g., tossing a coin has only one of two possible outcomes: heads or tails.

If the probability of the outcome true (p) is known, then all other parameters of the distribution, i.e., mean and variance, can be calculated. Hence, p is called the parameter of Bernoulli distribution.



Sample space $S = \{s,f\}$

Random variable X *defined by* $X(s) = 1, \quad X(f) = 0$

Distribution: $P(X = x) = p^x * (1-p)^{1-x}$, $x = 0, 1; 0 < p < 1$

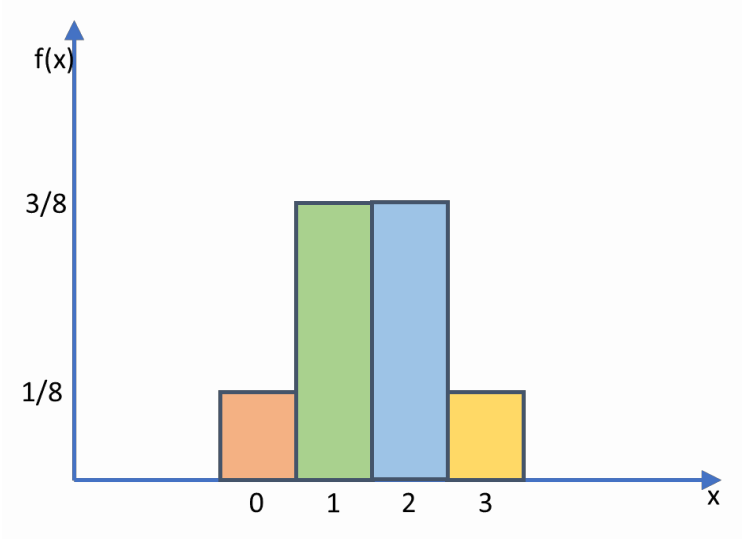
$\mu = p$

$\sigma^2 = p\,(1 - p)$

Binomial

When the Bernoulli distribution is repeated again and again, it is called binomial distribution. For example, if ten people visited my website, the probability that the first person subscribes to my newsletter is Bernoulli distribution. Similarly, the probability that the second person subscribes to my newsletter is Bernoulli distribution. But the probability that

five people out of ten subscribe to my newsletter is binomial distribution.



If the number of trials (n) executed during the experiment and the probability (p) of success are known, all other parameters of the distribution, i.e., mean and variance, can be calculated. Hence, **n** and **p** are called the parameters of binomial distribution.

$$P(x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

for x=0, 1, 2, ..., n

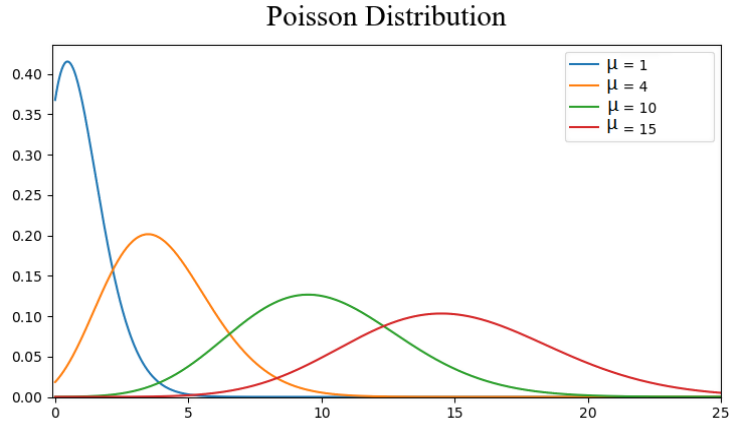
$$\mu = np$$

$$\sigma^2 = np(1-p)$$

$$\sigma = \sqrt{np(1-p)}$$

Poisson

Poisson is the probability distribution that applies to occurrences of an event over a specific interval. Mean and variance are the same in Poisson distribution. In this distribution, the assumptions are: the probability of success in an interval is same for all equal size intervals; and the probability of success is proportional to the size of the interval. This distribution is helpful in traffic planning, ATM refilling planning, etc. For example, if the number of trucks passing through the US–Mexican border is on average 96 per hour and the distribution is Poisson, then 192 trucks will cross the border in the next two hours.



The parameter of the distribution is mean (μ).

$$P(x) = \frac{e^{-\mu}\mu^{x}}{x!} \quad for \quad x = 0,\ 1,\ 2,\ldots$$

where μ is the mean number of successes in the interval

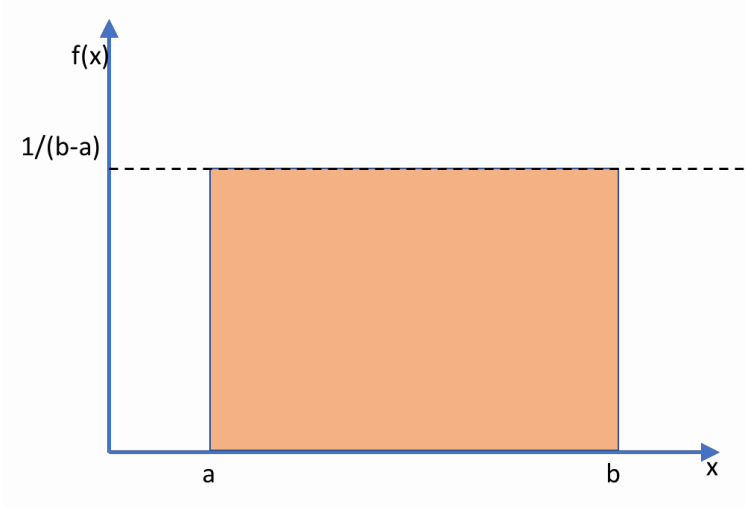
e – natural logarithm base

$E(X) = V(X) = \mu$

Continuous Probability Distributions

Uniform Distribution

When the probability of all possible values of a continuous random variable is the same then the distribution is called uniform distribution for a continuous random variable. In this distribution, an equal probability is assigned to all values between its minimum and maximum values.



When values a and b change, the length and breadth of the distribution will change, but the shape will remain same, i.e., it will be a rectangular shape.

Parameters of the distribution are a (Start) and b (End).

Probability density function: $f_X(x) = 1/(b-a)$, $a < x < b$

Mean, $\mu = (a + b)/2$

Variance, $\sigma^2 = (b - a)^2/12$

Gamma Distribution

Gamma is a family of distributions that are governed by the two parameters **α** and **λ**. A change in any of these values will result in a different distribution shape. Later, we will dive into a few specific graphs from the gamma family, e.g., exponential and Chi-square distributions.

In the example below, for blue graph line $\alpha = 2$ and $\lambda = 3$, and for red graph line $\alpha = 1$ and $\lambda = 4$. Gamma distribution is widely used in the insurance domain to predict claim amounts.

Probability density function: $f_X(x) = (\lambda^{\alpha} x^{\alpha-1} e^{-\lambda x})/\Gamma(\alpha)$, $x > 0$

Mean, $\mu = \alpha/\lambda$

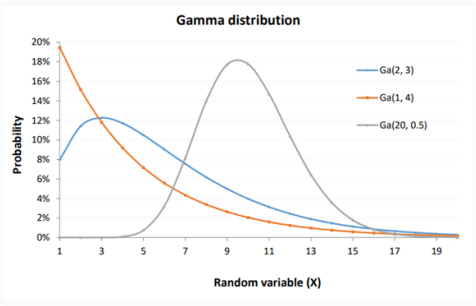
Variance, $\sigma^2 = \alpha/\lambda^2$

Special cases:

- Exponential distribution when $\alpha = 1$: $f_X(x) = \lambda e^{-\lambda x}$, $x > 0$
- *Chi-square* distribution with $\alpha = 2v$ (v any positive integer) and $\lambda = 1/2$

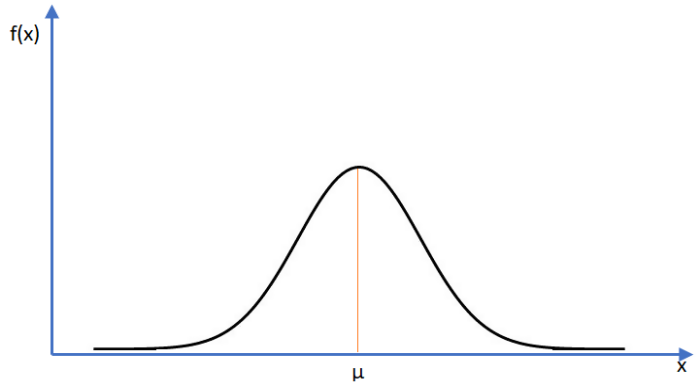
Plotting PDFs for different Gamma distributions using MS Excel.

X	Ga(2, 3)	Ga(1, 4)	Ga(20, 0.5)
1	7.96%	19.47%	0.00%
2	11.41%	15.16%	0.00%
3	12.26%	11.81%	0.00%
4	11.72%	9.20%	0.08%
5	10.49%	7.16%	0.75%
6	9.02%	5.58%	3.23%
7	7.54%	4.34%	8.17%
8	6.18%	3.38%	13.98%
9	4.98%	2.63%	17.73%
10	3.96%	2.05%	17.77%
11	3.12%	1.60%	14.71%
12	2.44%	1.24%	10.40%
13	1.90%	0.97%	6.44%
14	1.46%	0.75%	3.56%
15	1.12%	0.59%	1.79%
16	0.86%	0.46%	0.82%
17	0.65%	0.36%	0.35%
18	0.50%	0.28%	0.14%
19	0.37%	0.22%	0.05%
20	0.28%	0.17%	0.02%



Normal Distribution

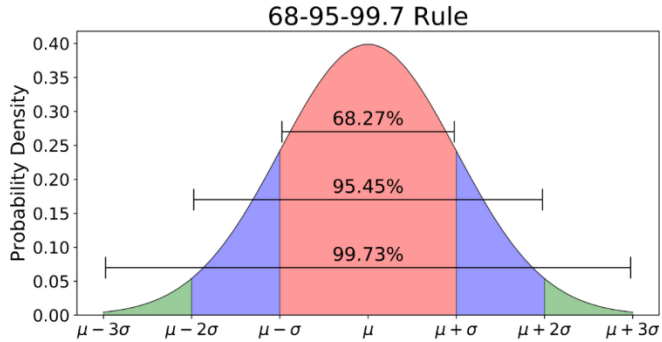
This distribution is bell shaped, and the distribution is symmetrical around the mean (**μ**). The spread of the distribution is decided by the standard deviation (**σ**). The higher the standard deviation, the higher the spread. If the standard deviation is low, then the distribution will shrink. The peak of the normally distributed curve is called *kurtosis*.



The probability density function of a normal random variable is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \qquad -\infty < x < \infty$$

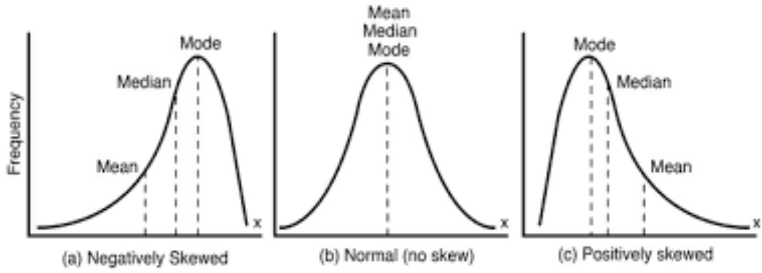
The distribution of the data points in a normal distribution is shown below. It shows that 68.27% of the data points will be between (μ - 1σ) and (μ + 1σ). Approximately 68% of the population lies within one standard deviation distance from the mean. Similarly, 95.45% of the population lies within two standard deviation distances from the mean. This means that the further the distance from the mean, the more population will be covered.



Skewness in the Distribution

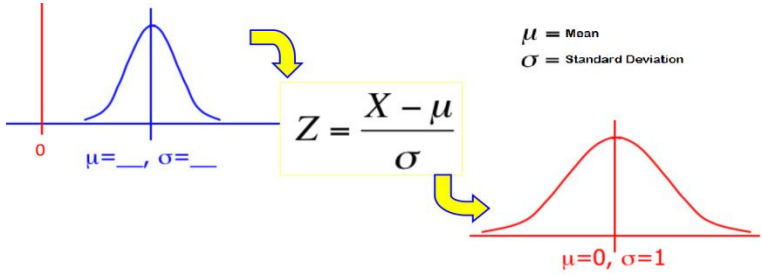
When a distribution is not symmetrical, it is called a skewed distribution. When most of the data points are on the right side of the distribution, it is considered negatively skewed. In negatively skewed distribution, the mode is greater than the mean.

When most of the data points are on the left side of the distribution, it is considered positively skewed. In positively skewed distribution, the mean is greater than the mode.



Standard Normal Distribution

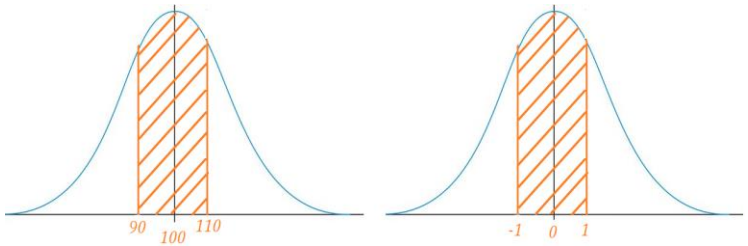
This is the standardized form of a normal distribution. All values in a population are brought to a single scale, e.g., a classroom has 20 students and for their age distribution, the mean = 25 and the standard deviation = 10. The formula below is used to standardize all values of x. For standard normal distribution, mean (μ) = 0 and standard deviation (σ) = 1.



Standard normal distribution helps us calculate the probability (the area under the curve). The probability between two points

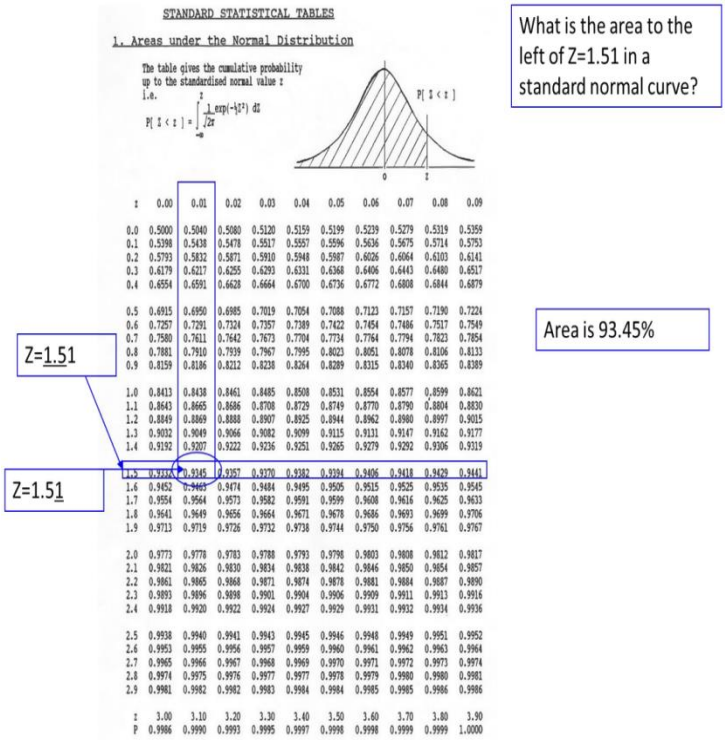
in a standard normal distribution is the same as the probability between corresponding points in a normal distribution. Probabilities for standard normal distribution are already available in a tabular format, which is known as the *Z table*.

Consider the following example. A normal distribution probability between 90 and 110, with a standard deviation of 10 and a mean of 100, will be same as the probability between ([90–100]/10 = -1) and ([110–100]/10 = 1), with a standard deviation of 1 and a mean of 0. In the figure below, the shaded region (area under the curve) is the same for both graphs.



Below the Z table is a pre-calculated probabilities table for standard normal distribution. Once the probabilities for standard normal distribution are known, they can be used in the normal distribution. Properties of the standard normal distribution are as follows:

1. Symmetrically distributed around mean (μ) = 0
2. Standard normal distribution = 1
3. The area under the curve of each side (left or right side of μ) = 0.5.



Population distribution – The distribution of all individual scores in the population.

Sample distribution – The distribution of all the scores in a sample.

Sampling distribution – The distribution of all possible sample means when taking samples of size n from the population. It is also called as distribution of the sample means.

Central Limit Theorem – If the sample size is sufficiently large (n = ~30), then the distribution of sample means will approximately follow a normal distribution, irrespective of whether the population distribution is normal or not.

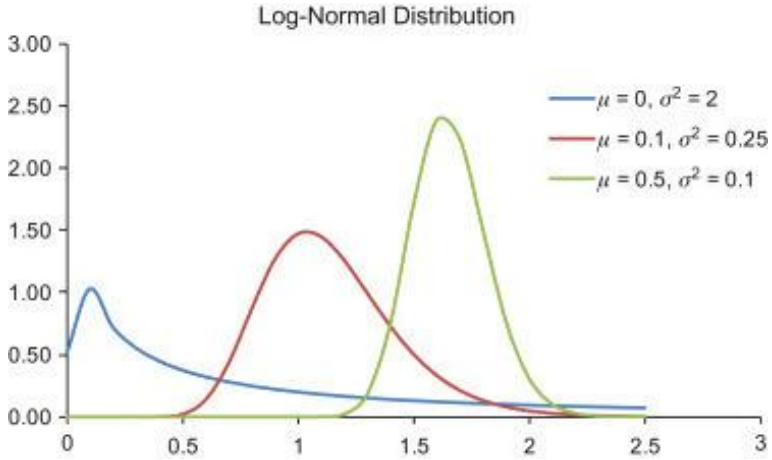
$$\mu_{\bar{x}} = \mu$$

$$\sigma_{\bar{x}}^2 = \sigma^2 / n \quad and \quad \sigma_{\bar{x}} = \sigma / \sqrt{n}$$

$\mu_{\bar{x}}$ is mean and $\sigma_{\bar{x}}$ is the standard error of sampling distribution.

Lognormal Distribution

When the logarithm of a continuous random variable is normally distributed, then the probability density function of the random variable is considered lognormally distributed.



Probability density function, mean, and variance of lognormal distribution are defined as:

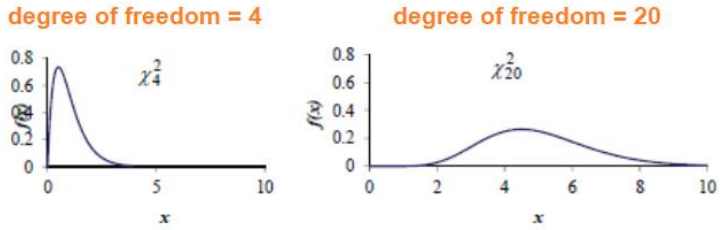
$$f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-\frac{(\ln x - \mu)^2}{\sqrt{2\sigma^2}}}$$

$$\mu^* = e^{\mu + \frac{\sigma^2}{2}}$$

$$\sigma^{*2} = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$$

Chi-square Distribution

Chi-square distribution is a family of distribution and a special case of Gamma distribution. The shape of the distribution is based on the degree of freedom and the sample mean. The higher the degree of freedom, the more it inches towards a normal distribution. A Chi-square test cannot have a negative random variable.



As we discussed earlier, it is often very costly to collect values from an entire population. Thus, we collect values for sample data and infer conclusions about the population from the sample.

It is extremely important to collect accurate samples. Otherwise, inferences made for the population will not be correct. For example, we want to find out the average age of a state. It is costly and time consuming to survey every person in the state. Instead, we select a few people as a sample and calculate the average age of this sample. In addition, we select

multiple samples to make the inferences about the whole population more generic. To make accurate assumptions about the population, the sample chosen should accurately represent the whole population.

In the above example, if we only surveyed the elderly or school-age people, for instance, then the inferences we make about the population will be skewed.

We will take another example to further understand population sampling and genericizing data.

A metrological department is studying ten UK cities to establish whether air pollution is at acceptable levels.

In this example, the selected ten cities are the sample, and all UK cities are the population.

Inferences made for the population from the sample are categorized into two types.

Estimation

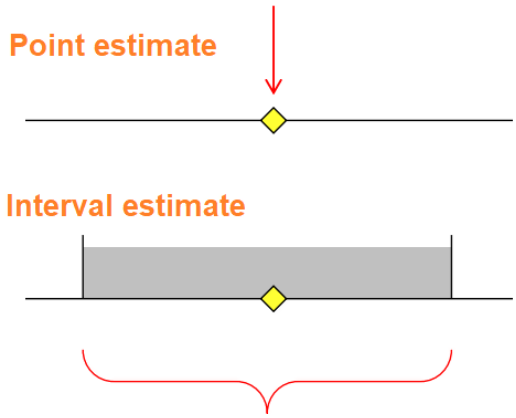
The objective of estimation is to uncover population parameters based on sample statistics, i.e., we use the sample mean to estimate the population mean.

Estimation is further categorized into two classes.

1. *Point estimation* – When the estimation is an exact value then it is called point estimation, e.g., the mean age of a college's students is estimated to be 21 years based on the sample mean of 50 students' ages. Estimating an exact value is difficult, and the chances of error are high.

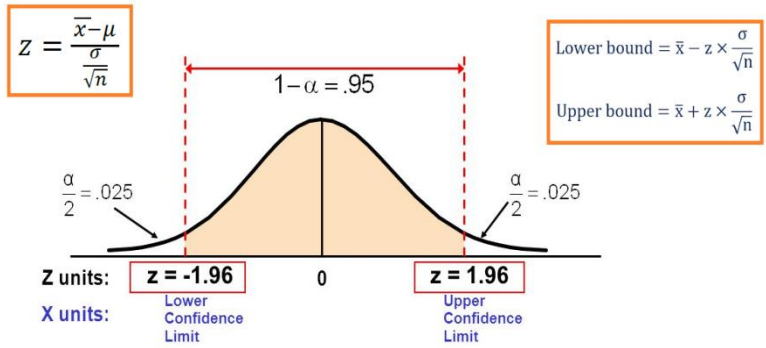
2. *Interval estimation–* When the estimation is a range or an interval of values then it is called interval estimation, e.g., the mean age of a college's students is estimated to be between 19 and 22 years based on the sample mean of 50 students' ages.

With interval estimation, a probability can be assigned to indicate the probability that the true value of the estimated population parameter will lie within the interval. This probability is called the **confidence level**.



Confidence Interval (1-α)

Confidence level relates to the level of confidence in the correctness of the estimate. Confidence level requirements differ from industry to industry. For example, a drug manufacturer's confidence level might be 99.99%, whereas a milk packaging dealer's confidence level can be as low as 95%.



The following presents a list of commonly used confidence levels:

Confidence Level

$1-\alpha$	α	$\alpha/2$	$z_{\alpha/2}$
.90	.10	.05	$z_{.05} = 1.645$
.95	.05	.025	$z_{.025} = 1.96$
.98	.02	.01	$z_{.01} = 2.33$
.99	.01	.005	$z_{.005} = 2.575$

Hypothesis Testing – "A hypothesis is a logical supposition, a reasonable guess, an educated conjecture. It provides a tentative explanation for a phenomenon under investigation."[v]

Let us try to understand the hypothesis with the help of an example. In a criminal trial, a jury has to decide between two hypotheses:

H_0: The defendant is innocent (null hypothesis)

H_1: The defendant is guilty (alternate hypothesis)

The jury does not know which hypothesis is correct. They must take a decision based on the evidence presented.

There are two possible decisions that can be made:

1. Conclude that there is enough evidence to support the alternative hypothesis. This means rejecting the null hypothesis in favor of the alternate hypothesis.
2. Conclude that there is not enough evidence to support the alternate hypothesis. This means *not* rejecting the null hypothesis in favor of the alternate hypothesis.

In hypothesis testing, there are two possible errors:

1. Type-I error occurs when we reject a true null hypothesis, e.g., when a jury convicts an innocent person.
2. Type-II error occurs when we do not reject a false null hypothesis, e.g., when a jury acquits a guilty defendant.

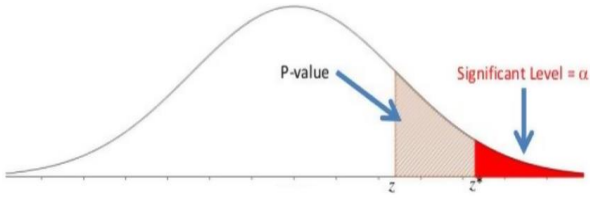
The probabilities of Type-I and Type-II errors are inversely related. Decreasing the probability of one increases the probability of others.

	Decision	
	Accept H_0	Reject H_0
H_0 (true)	Correct decision	Type I error (α error)
H_0 (false)	Type II error (β error)	Correct decision

There are two approaches to validating the null hypothesis.

P-value Test

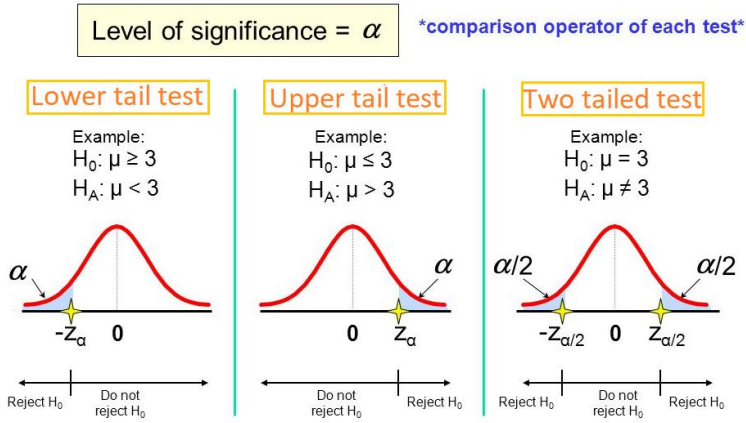
P-value testing determines the probability that your null hypothesis is correct. The null hypothesis will be rejected if the p-value is less than α (significance level). A higher p-value means stronger support for the null hypothesis. In the figure below, the region shaded in brown is the p-value, and the region shaded in red is the significance level. If the p-value region is lower than the significance level region, then we reject the null hypothesis.



Rejection Region

In the figure below, the blue shaded region is the rejection region. It illustrates that if the calculated z-test statistics is

laying below the shaded region, it rejects the null hypothesis. Here, α is the significance level for the null hypothesis.



There are six steps in hypothesis testing:

1. Define the null hypothesis and the alternate hypothesis.
2. Specify the significance level (i.e., determine the confidence level).
3. Select an appropriate statistical test.
4. Decide on the correct sampling distribution.
5. Randomly pick a sample from the population, and calculate test statistics.
6. Use p-value testing or rejection region to make a decision.

To this point, we have been discussing calculating the mean of the population when the standard deviation of the population is known. Now, let us consider how the standard deviation of

the population is calculated when the mean of the population is unkown. In real-life scenarios, the population standard deviation is not known, and the inferences need to be made from the sample. In such cases, the z-test cannot be used, but we can use the t-test to make deductions.

The t-test is used in two scenarios:

1. One sample t-test – This is used to compare a single sample to a population with a known mean but an unknown variance. The formula for t-statistics is like z-statistics, except that t-statistics uses the estimated standard error.

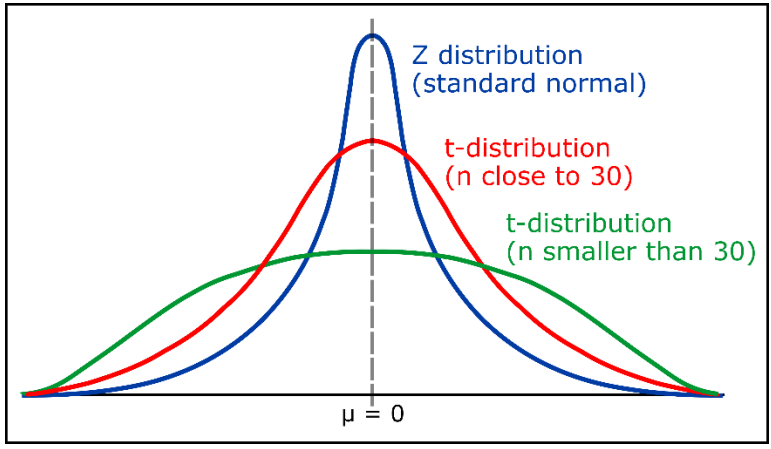
$Z = \frac{\overline{X} - \mu_{hyp}}{\sigma_{\overline{X}}}$	$t = \frac{\overline{X} - \mu_{hyp}}{s_{\overline{X}}}$
$\sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}}$	$s_{\overline{X}} = \frac{s}{\sqrt{n}}$
$\sigma = \sqrt{\frac{\Sigma(X-\mu)^2}{N}}$	$s = \sqrt{\frac{\sum(X-\overline{X})^2}{n-1}}$

In t-statistics, n-1 is used to calculate the standard deviation. The value n-1 is called the degree of freedom. The idea behind using n-1 is that when the mean of a sample is known, then the value of one element can be calculated using the remaining elements and the mean. So, if the mean is constant, there are only n-1 ways to manipulate the data. In simple terms, the degree of freedom is the number of scores in the sample that are subject to vary.

Let us put this in the context of an example. There is a bag with five balls labelled 1, 2, 3, 4, and 5. The mean of the number of balls is $(1 + 2 + 3 + 4 + 5)/5 = 3$. In this scenario, if four balls are out of the bag, we can calculate the number of balls still inside the bag by subtracting the mean (3) from the number of balls that are out of the bag (4).

Let us take another example. As the degree of freedom increases, the t-distribution approaches a normal distribution. The degree of freedom increases as the sample size increases and for large samples ($n > 30$), it almost becomes a z-distribution. As the degree of freedom increases, the uncertainty decreases.

The figure below shows how the distribution changes as the degree of freedom changes.



Steps for Hypothesis Testing

1. Formulate the hypothesis, e.g., the population mean is not equal to a specified value:

   $H_0: \mu = \mu_0$ $H_a: \mu \neq \mu_0$

2. Check the assumption:

- The sample is random.
- The population from which the sample is drawn is either normal or the sample size is large.

3. Calculate the test statistics:

$$t = \frac{\bar{x} - \mu}{s_{\bar{x}}} \quad \text{where } s_{\bar{x}} = \frac{s}{\sqrt{n}}$$

where

μ = Proposed constant for the population mean

$\bar{x}$ = Sample mean

n = Sample size (i.e., number of observations)

s = Sample standard deviation

$s_{\bar{x}}$ = Estimated standard error of the mean (s/sqrt(n))

4. Calculate the p-value based on an appropriate alternative hypothesis.

5. Draw a conclusion.

Two sample t-test – This test is used to determine whether the mean of one group is equal to, larger than, or smaller than the mean of another group, e.g., is the mean cholesterol of people taking drug A lower than the mean cholesterol of people taking drug B.

In this scenario, samples are taken from two different populations (people taking drug A and people taking drug B). Based on these samples, we try to infer weather the two populations to which these samples belong to are the same or not.

Steps for Hypothesis Testing

1. Formulate the hypothesis, e.g., the population means of two groups are not equal:

   $H_0: \mu_1 = \mu_2$ $\qquad\qquad$ $H_a: \mu_1 \neq \mu_2$

2. Check the assumptions:

- The two samples are random and independent.
- The populations from which the samples are drawn are either normal or the sample sizes are large.
- The populations have the same standard deviation.

3. Calculate the test statistics:

$$t = \frac{(\overline{x_1} - \overline{x_2}) - (\mu_1 - \mu_2)}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Where

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

4. Calculate p-values based on an appropriate alternative hypothesis.

5. Draw a conclusion.

Paired t-test – The paired t-test is used to compare the means of two dependent samples. This is called a paired t-test because subjects are the same in both the samples, but the environment is different.

For example, a researcher would like to determine if background noise causes people to take longer to complete math problems. The researcher gives 20 subjects two math tests in two different environemnts—one with complete silence and one with background noise—and records the amount of time each subject takes to complete each test. In this scenario, test scores are compared regardless of background noise.

Steps for hypothesis testing:

1. Formulate the hypothesis, e.g., a population mean difference is not zero:

$\mathbf{H_0: \mu_{difference} = 0}$ $\mathbf{H_a: \mu_{difference} \neq 0}$

2. Check the assumptions:

- The sample is random.
- The data is a matched pair.
- The differences have a normal distribution or sample size.

3. Calculate the test statistics:

$$t = \frac{\bar{d} - 0}{s_d / \sqrt{n}}$$

Where

$\bar{d}$ is the mean of the differences

s_d is the standard deviation of the differences

4. Calculate p-values.
5. Draw a conclusion.

We have discussed the sample mean and population mean, and how we can approximate population mean based on a sample mean when sample size is large or small or the standard deviation of the population is known or not. The next thing we will look at is sample variance. The sample variance is a random variable, which we can approximate to calculate the population variance. Or, if we have two populations, we can find out whether they have the same variance.

Chi-square distribution: This is denoted by the Greek letter *chi* (χ). It can have many degrees of freedom. The Chi-square distribution is a sum of squares, which means that it cannot be negative.

Here is the Chi-square distribution with 1 degree of freedom:[vi]

$$z = \frac{(X - \bar{X})}{SD}; z = \frac{(X - \mu)}{\sigma} \longrightarrow \text{z score}$$

$$z^2 = \frac{(X - \mu)^2}{\sigma^2} \longrightarrow \text{z score squared}$$

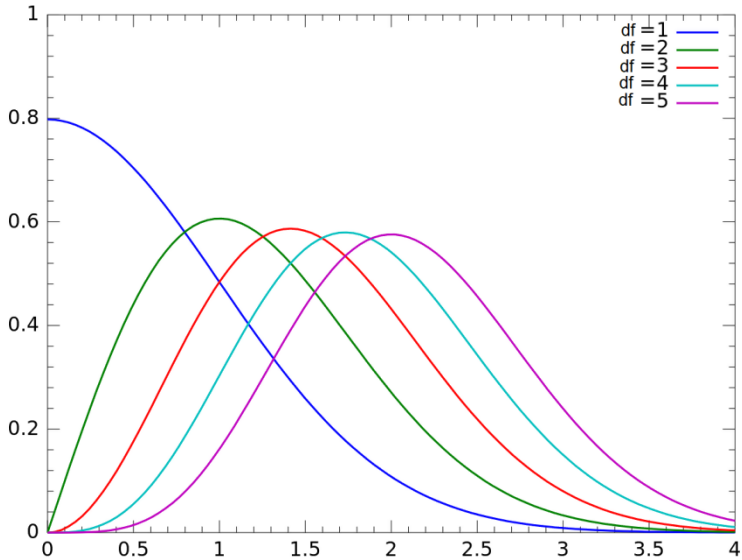
$$z^2 = \chi^2_{(1)} \longrightarrow \text{Make it Greek}$$

Here is the Chi-square with two degrees of freedom:

$$z_1^2 = \frac{(X_1 - \mu)^2}{\sigma^2}; z_2^2 = \frac{(X_2 - \mu)^2}{\sigma^2}$$

$$\chi_{(2)}^2 = \frac{(X_1 - \mu)^2}{\sigma^2} + \frac{(X_2 - \mu)^2}{\sigma^2} = z_1^2 + z_2^2$$

The distribution of a Chi-square depends on one parameter: the degree of freedom (df). As the df increases, the curve is less skewed and more normal.



The characteristics of Chi-square distribution are as follows:

1. The expected value of Chi-square distribution is its degrees of freedom.
2. The mean of Chi-square distribution is its degree of freedom.

3. The expected variance of the Chi-square distribution is 2*df.

4. Chi-square is additive.

$$\chi^2_{(v_1+v_2)} = \chi^2_{(v_1)} + \chi^2_{(v_2)}$$

Distribution of sample variance – The sample variance is a random variable distributed as Chi-square with n-1 degrees of freedom.

$$\chi^2_{(N-1)} = \frac{(N-1)s^2}{\sigma^2}$$

Where

$$s^2 = \frac{\sum (y-\bar{y})^2}{N-1}$$

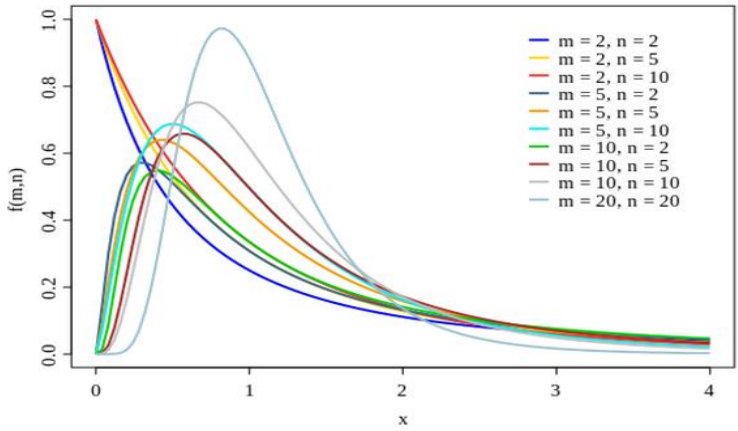
We can use the information about the sampling distribution of the variance estimate to find the confidence intervals and conduct statistical tests.

Confidence intervals for the variance. For 95% confidence level it can be represented as :

$$p\left[\frac{(N-1)s^2}{\chi^2_{(N-1;.025)}} \le \sigma^2 \le \frac{(N-1)s^2}{\chi^2_{(N-1;.975)}}\right] = .95$$

If the degree of freedom is lower, then it will not be a symmetric distribution. Hence, in a two-tail test, separate p-values need to be calculated at both ends.

F-distribution: This is a family of curves, and each curve is defined by two degrees of freedom. F-distributions are positively-skewed distributions. An f-distribution is used to compare the variance of two samples and approximate it to calculate population variance.



So far, we have discussed four distributions: z-test, t-test, Chi-square, and f-test. Out of these four distributions, only the z-test has no degrees of freedom associated with it. The other three distributions are associated with degrees of freedom. F has two degrees of freedom associated with it; z and t are closely related to the sampling distribution of the means. Z-distribution is used when population standard deviation is known; t-distribution is used when population standard deviation is unknown, and the sample standard deviation is used to approximate population standard deviation. Chi-square and f-distributions are closely related to the sampling distribution of variances.

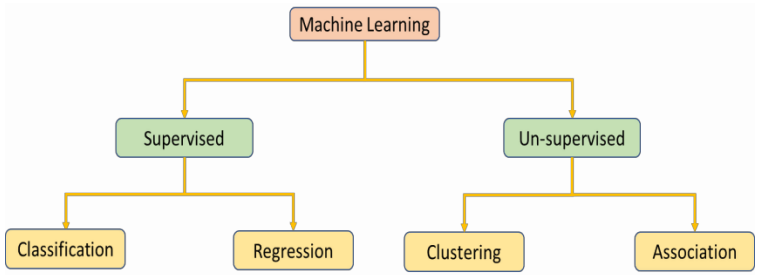
Machine learning and deep learning algorithms use probability distribution functions and statistics for predictions and classifications.

Probability distribution function and statistics are the backbone of machine learning. Many of the above statistical approaches are used in exploratory data analysis and feature engineering. For example, if we want to know how a house with three bedrooms is different from a house with four or more bedrooms, we can use statistical approaches to establish relationships within the same feature or with other features.

Chapter 5: Machine Learning Algorithms

In the previous chapters, we discussed the tools that help us understand machine learning algorithms and build machine learning models. In this chapter, we will learn about various machine learning algorithms in practice, which are commonly used for prediction, classification, clustering, etc.

Machine learning is broadly classified into supervised and unsupervised learnings. Supervised learning means that the algorithms are supervised during the training phase. In other words, in order to train these algorithms, we need data that have labeled targets, e.g., if a model is being created for predicting house prices then the historical data that is used to train the model should have a target column stating the price of a house.

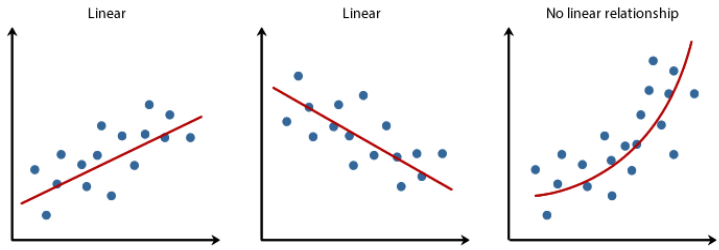


Supervised machine learning problems are broadly classified into two categories: regression and classification.

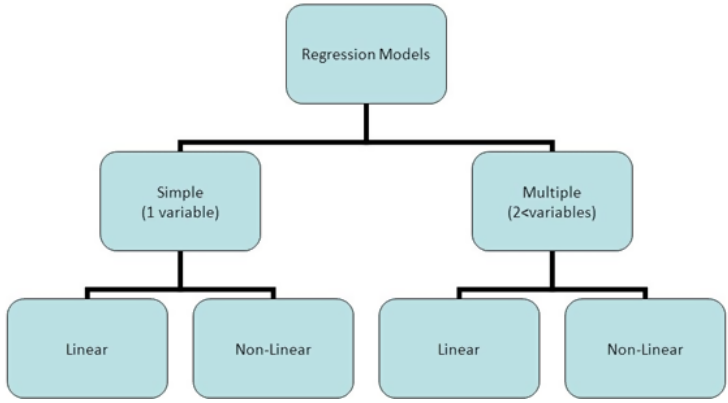
Regression

We use this form of predictive modeling technique to identify a relationship between a dependent variable (target) and independent variables (predictors/features). In regression

analysis, the target is always a continuous variable, and the predictors can be continuous or discrete in nature. Regression is best used for finding causal effect relationships between the variables, forecasting, time series modeling, etc. In regression analysis, the model tries to fit a curve to the data points in such a manner that the difference between the data point and the curve is at a minimum.



Types of regression models:



When the dependent variable (target) is dependent on only one independent variable (predictor/feature), then it is called simple regression. If the target is dependent on multiple predictors, then it is called multiple regression.

In regression analysis, labeled historical data is available. Machine learning algorithms collect insights from this data and use these insights to predict a dependent variable (target) for new instances of an independent variable (e.g., predicting a house price in the United States, predicting marketing expenditure to boost sales of a certain product, etc.)

Classification

These algorithms are used to predict the target that has discrete values (known as target classes, e.g., orange, pineapple, and lime are different classes of fruits). For this kind of use, machine learning algorithms collect insights from the historical labeled data, and use these insights to predict the target class (e.g., predicting a wine quality classification, predicting whether a patient is suffering from cancer or not, etc.).

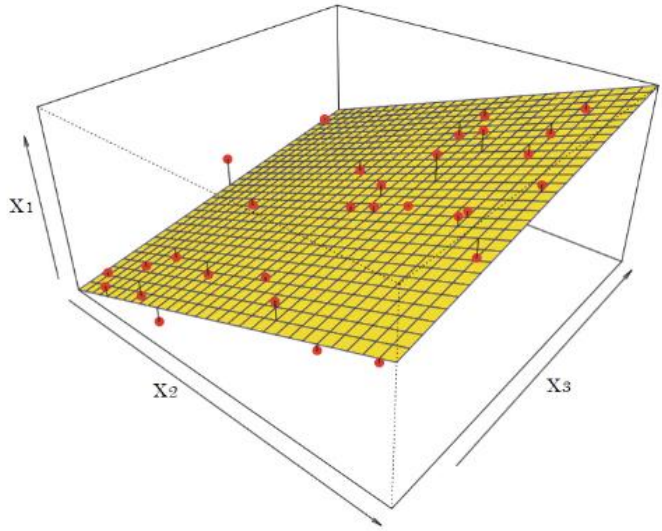
Commonly used regression machine learning algorithms are linear regression, decision trees, support vector machines (SVMs), naïve bayes, nearest neighbor, neural network, etc. Let us discuss these algorithms in detail to build an understanding of how to use them and their pros and cons.

Linear Regression

Linear regression is one of the widely known machine learning algorithms for regression. This algorithm is mainly used to solve regression use cases where the dependent variable is a linear function of independent variables. i.e., for predicting a value for the target that is continuous in nature and can be represented as a linear function of predictors/features, e.g., predicting the rental rates for a bike renting company, where bike rentals can be represented as a linear function of weather, wind speed, etc.

A linear regression model establishes a relationship between the dependent variable (target) and independent variable (in case of a single feature) using a best-fit straight line, also known as the regression line. The linear regression model tries to fit all data points along a straight line and minimizes the distance between the straight line and the data point to get the best fitted straight line.

For one independent variable, the predicted curve is a line; for two independent variables, the predicted curve is a plane; and for more than two independent variables, the predicted curve is a hyperplane.

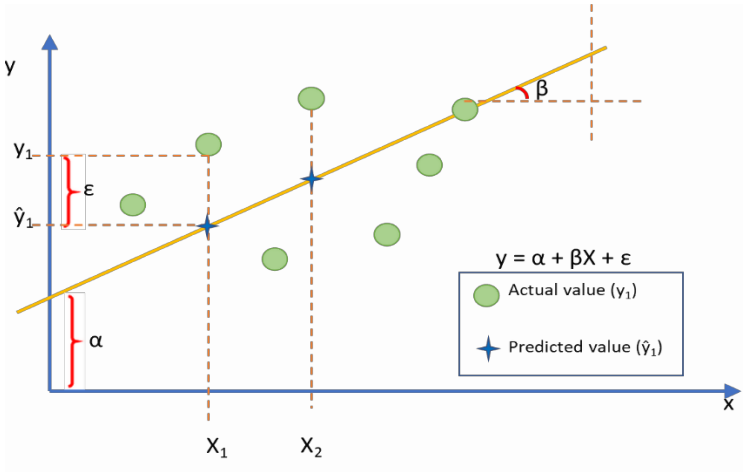


Simple Linear Regression

When the dependent variable (target) is dependent on only one independent variable (predictor), then the linear regression model is called a simple linear regression model and is represented as:

$$y = \alpha + \beta X + \varepsilon$$

where y is the dependent variable, X is the independent variable, α is the intercept, β is the slope of the line, and ε is the error term (the difference between the actual value and predicted value). This equation can be used to predict y for a given value of X.



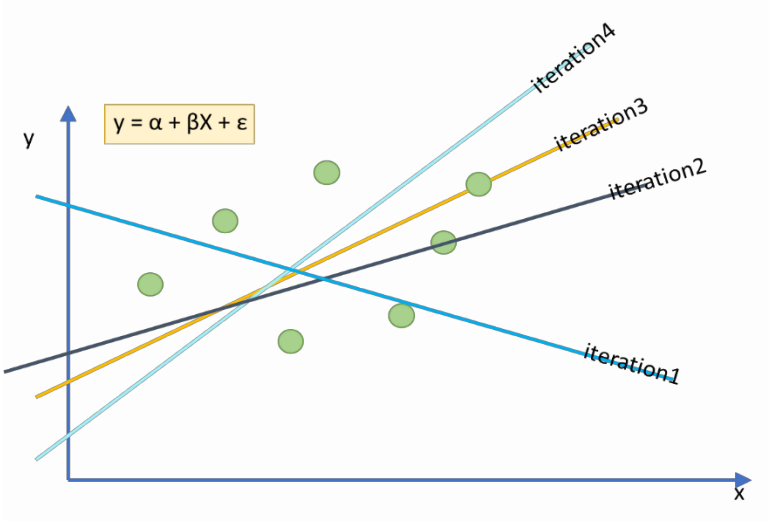
As shown in the above image, green circles are original data points, and the orange line is the fitted line (all predicted values of y lies on this line). A linear regression algorithm modifies α, β, and ε to find a best-fit regression line so that the error between the predicted value and the actual value is minimum.

α and β are called model coefficients, and the machine learning algorithm learns these coefficients from training data by finding out the relationship between the dependent variable (y) and the independent variable (X). Once these model coefficients are learned by the machine learning model, the model can be used to predict the target variable (y).

Finding the best fitted regression line is an iterative process. With each iteration, the algorithm calculates the mean squared error (MSE), and in the next iteration, the algorithm updates model parameters to shift the line from the previous position to reduce the mean squared error.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

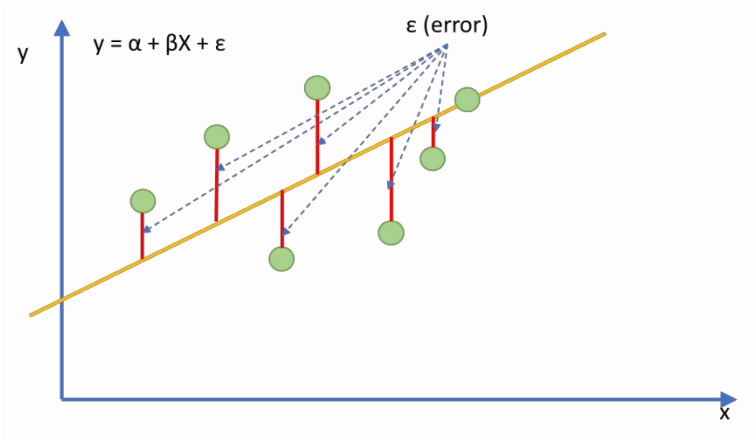
* n is the number of data points
* Y_i represents observed values
* $\hat{Y}_i$ represents predicted values



The error in prediction for each observation is represented as shown in the image below. In the graph below, the red lines are errors (ε), i.e., the difference between the predicted and actual value of the target.

$$\varepsilon = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)$$

* n is the number of data points
* Y_i represents observed values
* $\hat{Y}_i$ represents predicted values



For the points above the fitted line, ε will be positive, and for the points below the fitted line, ε will be negative. Hence, if all the errors are added, then there are few possibilities of getting positive and negative errors canceling each other out. To avoid this, in general practice squared error is used and it is calculated by squaring the error for each observation and then summing them up.

Multi Linear Regression

When the dependent variable (y) is dependent on more than one independent variable, then the regression model is called a multi linear regression model and is mathematically represented as:

$$y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots\ldots + \beta_n X_n + \varepsilon$$

$$y = \alpha + \sum \beta_i X_i + \varepsilon \qquad i = 1,2,3, \ldots\ldots, n$$

where **n** is the number of independent variables (features), **y** is the dependent variable, X_1, X_2, X_3…, X_n are independent

variables, α is the y intercept, $\beta_1, \beta_2, \beta_3, \ldots\ldots\ldots \beta_n$ are slopes with respect to corresponding independent variables, and ε is the error between the actual target value and predicted target value.

Alternatively, the predicted variable (y) is the weighted sum of independent variables (X_i) where model coefficients are the weights for each independent variable.

The linear regression model function can be rearranged and written as follows to solve the linear function for multiple observations in the dataset. Where $X_{1,1}, X_{1,2}, X_{1,3}, \ldots\ldots, X_{1,n}$ is one observation in the dataset.

$$\begin{pmatrix} 1 & X_{11} & X_{21} & X_{31} & \ldots\ldots\ldots & X_{n1} \\ 1 & X_{12} & X_{22} & X_{32} & \ldots\ldots\ldots & X_{n2} \\ 1 & X_{13} & X_{23} & X_{33} & \ldots\ldots\ldots & X_{n3} \\ & \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{1m} & X_{2m} & X_{3m} & \ldots\ldots\ldots & X_{nm} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix}$$

The above linear regression model is also known as the ordinary least squares (OLS) model since the model uses the technique of minimizing the mean of squared errors. For a good linear model, the MSE should be normally distributed. If the MSE distribution is skewed, then there are opportunities to improve the accuracy.

Mean squared error is the most commonly used error calculation method. However, other methods are also available for error calculation. Some of the commonly used error methods are:

Mean Absolute Error	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert(Y_i - \hat{Y}_i)\rvert$	$*\ n$ is the number of data points $*\ Y_i$ represents observed values $*\ \hat{Y}_i$ represents predicted values
Root Mean Squared Error	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$	
Mean Squared Error	$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$	
Mean Absolute Percentage Error	$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{(Y_i - \hat{Y}_i)}{Y_i}$	

When there are more features, OLS linear regression models tend to overfit (the model becomes more specific to the training dataset instead of being a generic solution). Hence, the prediction accuracy of the test dataset is significantly low compared with the prediction accuracy of the training dataset. To overcome this situation, a penalty term is added to the regression model. This is also called the linear model regularization.

These penalties are known as L1 and L2 penalties. A linear regression model with an L1 penalty is known as a lasso regression model, and a linear regression model with an L2 penalty is known as a ridge regression model.

For I = 1, 2, 3, ……., n

$y = \alpha + \sum \beta_i X_i + \lambda \sum |\beta_i|$ (Lasso Regression Model – L1)

$y = \alpha + \sum \beta_i X_i + \lambda \sum \beta^2_i$ (Ridge Regression Model – L2)

In the above equations, λ is the regularization parameter, $\sum|\beta_i|$ is the lasso penalty (L1), and $\sum \beta^2_i$ is the ridge penalty (L2).

Regularization makes the model more generic by making sure that all model coefficients have almost the same values so that each feature will have almost the same impact on the target for the same change in the independent variable. With regularization, the model becomes more generic, but the accuracy of the model reduces because now more important features have less impact on the target variable. So, there is a trade-off between accuracy and generalization. This trade-off can be controlled via the regularization parameter λ.

The regularization parameter (λ) and the penalty term (L1 or L2) are the only hyperparameters for linear regression models.

Linear regression models are easy to understand and perform well for a sparse dataset. These models are quick to learn and predict. If the dataset has highly corelated features, then linear regression models do not perform well.

Linear Regression Assumptions

Linear regression models work on the following assumptions:

1. Linear relationship – The relationship between the independent variable and the dependent variable is linear. This relationship can be checked by plotting the dependent variable against the independent variable. Linear models are highly sensitive to outliers; hence, the dataset should be cleaned properly to handle outliers and missing data.

2. Multivariate normality – All independent variables should be normally distributed. To address the problem, non-linear transformation lognormal can be used.

3. No or little collinearity – Independent variables must not be highly corelated to each other. To avoid this scenario, highly correlated features should be removed from the dataset.

4. No auto-correlation – Error terms (residuals) should not be correlated with each other. This means one error term should not be dependent on another error term.

5. Homoscedasticity – Error terms (residuals) should be equally distributed on both sides of the regression line. A scatter plot might help to identify homoscedasticity in the data.

Benefits of Linear Regression

1. Easy to interpret

2. Unambiguous and fast estimation

Disadvantages of Linear Regression

1. Only mean values are estimated.

2. Errors (residuals) should follow a normal distribution.

3. The relationship between dependent and independent variables is linear.

Examples

Regression models are used for house-price prediction, oil-price prediction, economic growth prediction, salary estimation, etc.

Logistic Regression

One assumption in the linear regression model is that the MSE follows a normal distribution. This assumption fails when the dependent variable is a categorical variable. So, for categorical dependent variables, linear regression cannot be used effectively. In this section, we will discuss a regression model to predict dependent variables that are categorical in nature. For example, we can use logistic regression to determine whether a patient will respond to a certain medical procedure or not, whether a plant will survive or not, for weather forecasting (predicting temperature is a regression problem since temperature is a continuous variable, but predicting if it is going to rain tomorrow or not is a classification problem since the dependent variable is categorical—it will either rain or not), etc.

For a dependent categorical variable, we could set up a linear regression model to predict individual category memberships if numeric values (e.g., 0 and 1) are used to represent the two categories. This model does not work well for a few reasons. First, the categorical values (0 and 1) are arbitrary, so modeling the actual values of the dependent variable is not exactly of interest, e.g., the model predicts the value as 0.6, but the only meaningful values for the dependent variable are 0 or 1.

Second, the probability that each observation in the dataset will fall into any of the categories is problematic, e.g., the probability of rain is higher if there is more moisture in the air. Thus, as the level of moisture in the air increases, the probability of rain increases.

Hence, it is better to model for predicting probabilities of the dependent variable. But, there is the constraint that probability cannot be less than zero or greater than 1, e.g., the probability of rain tomorrow cannot be -0.3 or 1.2; it should be between 0 and 1. Logistic regression avoids this situation by expressing predictions in terms of odds rather than probabilities. The odds $\frac{P}{(1-P)}$ in favor of an event is the ratio of probability that it will occur to the probability that it will not occur. The odds of the success represent the same information as that of the probability of success, but on a different scale. Probability is between 0 and 1; 0.5 being in the middle. Odds are between $-\infty$ and ∞; 1 being in the middle.

It is better to assume the relationship between dependent and independent variables as sigmoidal (S-shaped) instead of linear. Other functions are also available, e.g., hyperbolic tangent (tanh) to achieve a linear relationship between dependent and independent variables, but a sigmoid relationship results in some nice simplifications.

A linear relationship between the probability of the dependent variable and the independent variables can be established using various functions. However, the logit function is most commonly used. This linear relationship can be represented as:

$$\ln\left(\frac{P}{(1-P)}\right) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots\ldots + \beta_n X_n + \varepsilon$$

where ln is the natural logarithm.

This way of expressing probability results in a linear function of independent variables.

$$\ln\left(\frac{P}{(1-P)}\right) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots\ldots + \beta_n X_n + \varepsilon$$

$$S = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots\ldots + \beta_n X_n + \varepsilon$$

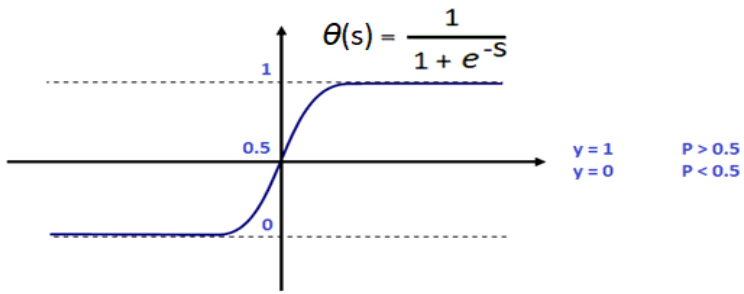
$$\ln\left(\frac{P}{(1-P)}\right) = S = \alpha + \sum_{i=1}^{n} \beta_i X_i$$

$$\frac{P}{(1-P)} = e^S$$

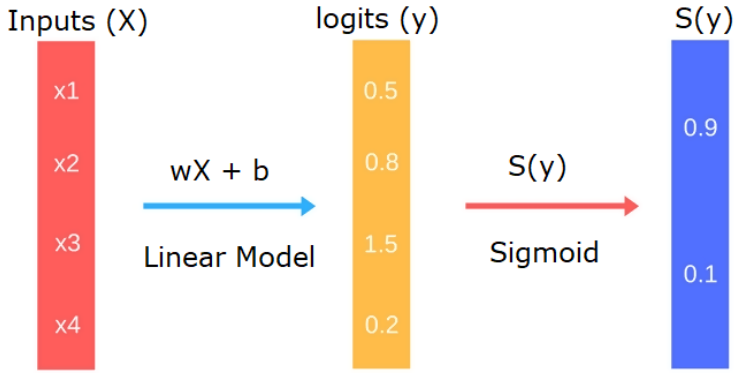
$$P = \frac{e^S}{1+e^S}$$

$$\theta(s) = P = \frac{1}{1+e^{-S}}$$

The probability of an event occurring is P and not occurring is 1-P. However, the logistic regression method calculates the category of the dependent variable based on the probability threshold. By default, the threshold is set to 0.5, which means that if the predicted probability is greater than 0.5 then the category is predicted as 1; otherwise, it is 0.



The logistics regression model steps are depicted below.



With this, we want to learn a hypothesis (h(X)) that best fits the above steps according to some error function:

$$h(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x}) \approx f(\mathbf{x})$$

Here, W^T is the transposed vector of β.

The probability of y given x can be represented as:

$$P(y \mid \mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{for } y = +1 \\ 1 - h(\mathbf{x}) & \text{for } y = -1 \end{cases}$$

if $y = +1$ then $h(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x}) = \theta(y\ \mathbf{w}^T\mathbf{x})$

if $y = -1$ then $1 - h(\mathbf{x}) = 1 - \theta(\mathbf{w}^T\mathbf{x}) = \theta(-\mathbf{w}^T\mathbf{x}) = \theta(y\ \mathbf{w}^T\mathbf{x})$

The likelihood is defined for a dataset D with N samples given a hypothesis (denoted arbitrarily as g here). Likelihood is an informal way of discussing the likeliness that something will happen, without specific references to numerical probability.[vii]

$$L = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i)^{1-y_i}$$

Here, x_i is the vector of features, y_i is the observed class, and the probability is p if $y_i = 1$, or (1- p) if $y_i = 0$.

$$L(\mathcal{D} \mid g) = \prod_{n=1}^{N} P(y_n \mid \mathbf{x}_n) = \prod_{n=1}^{N} \theta(y_n \ \mathbf{w}_g^T \mathbf{x}_n)$$

Now, finding a good hypothesis is a matter of finding a hypothesis parameterization that maximizes the likelihood.

$$\mathbf{w}_h = \underset{\mathbf{w}}{\operatorname{argmax}} \ L(\mathcal{D} \mid h) = \underset{\mathbf{w}}{\operatorname{argmax}} \ \theta(y_n \ \mathbf{w}^T \mathbf{x}_n)$$

Maximizing the likelihood is equivalent to maximizing the logarithm of the function since the natural logarithm is a monotonically increasing function:

$$\underset{\mathbf{w}}{\operatorname{argmax}} \ \ln\left(\prod_{n=1}^{N} \theta(y_n \ \mathbf{w}^T \mathbf{x}_n)\right)$$

We can maximize the above, proportional to a constant $1/N$:

$$\underset{\mathbf{w}}{\operatorname{argmax}} \ \frac{1}{N}\ln\left(\prod_{n=1}^{N} \theta(y_n \ \mathbf{w}^T \mathbf{x}_n)\right)$$

Now maximizing it is the same as minimizing its negative:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \left[-\frac{1}{N}\ln\left(\prod_{n=1}^{N} \theta(y_n \ \mathbf{w}^T \mathbf{x}_n)\right)\right]$$

It can be rearranged as:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \ \frac{1}{N}\sum_{n=1}^{N} \ln\left(\frac{1}{\theta(y_n \ \mathbf{w}^T\mathbf{x}_n)}\right)$$

Expanding the logistic function:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \ \frac{1}{N}\sum_{n=1}^{N} \ln\left(1+e^{y_n \ \mathbf{w}^T\mathbf{x}_n}\right)$$

Now we have a much nicer form for the error measure, known as the cross-entropy error:

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1+e^{-y_n \ \mathbf{w}^T\mathbf{x}_n}\right)$$

So, to learn a hypothesis, we want to perform the following optimization:

$$\mathbf{w}_h = \underset{\mathbf{w}}{\operatorname{argmin}} \ E_{\text{in}}(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \ \frac{1}{N}\sum_{n=1}^{N} \ln\left(1+e^{y_n \ \mathbf{w}^T\mathbf{x}_n}\right)$$

The learning algorithm is how we search the set of possible hypotheses (hypothesis space H) for the best parameterization (in this case weight vector w). This search is an optimization problem looking for the hypothesis that optimizes error (cross-entropy).

Cross-entropy function is a convex function. Hence, it will have only one minimum, also known as the global minimum. To reach the global minimum, we will use batch gradient descent, which calculates gradient from all data points.

Gradient descent is a general method and requires twice the differentiability for smoothness. It updates the parameters using a first order approximation of the error surface.

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \eta \nabla E_{\text{in}}(\mathbf{w}_i)$$

Here, the model coefficient vector W_{i+1} in the current iteration is the sum of model coefficients of previous iterations (W_i) and negative of the partial derivative of model coefficients of previous iterations ($\nabla E_{in}(W_i)$) multiplied by learning rate (η).

$$\nabla E_{\text{in}}(\mathbf{w}_i) = -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}}$$

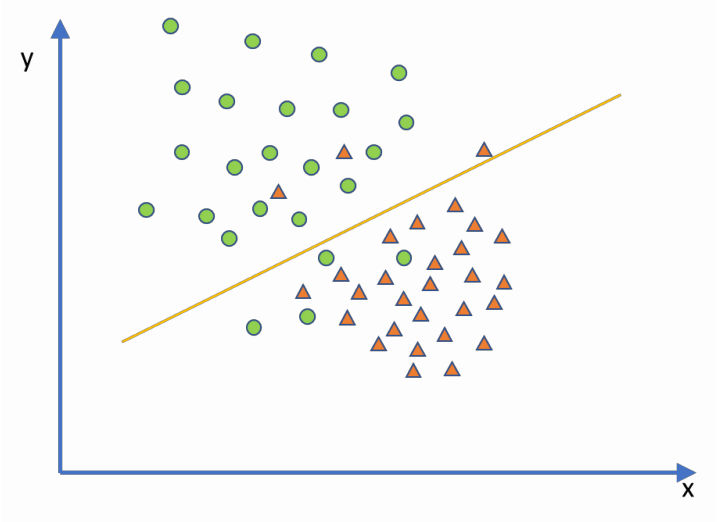
$$\mathbf{w}_{i+1} = \mathbf{w}_i + \eta \left(\frac{1}{N}\sum_{n=1}^{N}\frac{y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}_i^T \mathbf{x}_n}}\right)$$

Because logistics regression predicts probabilities, rather than just classes, it can be fitted using likelihood.

Maximum likelihood estimation: The goal of maximum likelihood estimation is to find an optimal way to fit a distribution to the data. It is a method that determines values for the parameters of a model. Parameter values are selected to maximize the likelihood that the process described by the model produces the data that were observed.

It is important to note that the training dataset does not provide a probability of the class; rather it describes the class itself.

Since the regression method is applied to predict probabilities of the dependent variable, this algorithm is called the logistic regression; however, it is a classification model. The output of the logistic regression model is a linear line separating the area classified as Class 1 (●) from the area classified as Class 2 (▲).



In other words, any data point that lies above the orange line is classified as Class 1 and the data points that lie below the orange line are classified as Class 2.

Logistic regression can be binomial or multinomial. Binomial logistic regression deals with the situations where the dependent variable has only two possible outcomes, like the tossing of a coin. In binomial logistic regression, the dependent variable is coded as 0 or 1. If the outcome is a success, then it is 1; otherwise, it is 0.

For binomial logistic regression, the accuracy is measured using ROC-AUC, where ROC stands for receiver operating characteristics, and AUC stands for area under the curve.

Multinomial logistic regression deals with the situations where the dependent variable can have more than two outcomes, e.g., wine quality classification (Class 1, Class 2, Class 3, etc.). It works on the concept of one-vs-rest. In the one-vs-rest approach, a binary model is learned for each class, which tries to separate this class from all other classes, resulting in as many binomial models as there are classes. As such, if the outcome can have four classes, then the multinomial logistic regression will create four binomial models to determine the classes, i.e., equation # 1 for Class 1 or not Class 1, equation # 2 for Class 2 or not Class 2, equation # 3 for Class 3 or not Class 3, and equation # 4 for Class 4 or not Class 4.

To make a prediction, all these binomial models run and the classifier that has the highest score wins and this class is returned as a prediction.

Since logistic regression also uses a linear function to imply regression, L1 and L2 penalties can be applied to reduce overfitting and make the model more generalized. Along with L1 and L2 penalties, the regularization parameter can be applied to control the amount of penalties.

When the number of independent variables is more, a linear model for classification becomes very powerful and guarding against overfitting becomes increasingly important.

The main parameter of the logistic regression model is the regularization parameter. This parameter should be tuned to balance the trade-off between overfitting and generalization. Another important parameter is the penalty. Either the L1 or

L2 penalty can be implemented for better results. The L1 penalty is more useful for a model explanation because it uses only important features for model creation.

Benefits of Logistic Regression

1. Probabilities can be effectively modeled as a function of independent variables.
2. There is no need to worry about violation of ordinary least square (OLS) assumptions.
3. It does not require independent variables to be normally distributed.
4. The prediction falls between 0 and 1.

Disadvantages of Logistic Regression

1. It forfeits the simple interpretation of linear coefficients.
2. R-squared cannot be used as an accuracy measure.
3. Accuracy is low for small datasets.

Examples

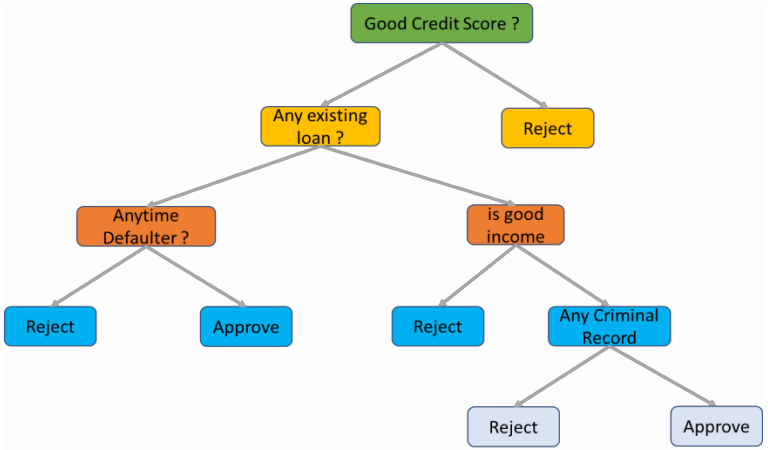
Logistic regression models are used for handwriting recognition, bank loan default predictions, bank loan approval predictions, credit card approval predictions, etc.

Decision Trees and Random Forest

A decision tree is essentially a series of if-else conditions that lead to a decision. Decision trees are widely used models for classification and regression use cases.

A decision tree is a type of supervised learning algorithm (having a predefined target variable). It works for both categorical and continuous input and output variables. In this technique, the dataset is split into two or more homogeneous sets (or sub-populations) based on the most significant splitter/differentiator in input variables.[viii]

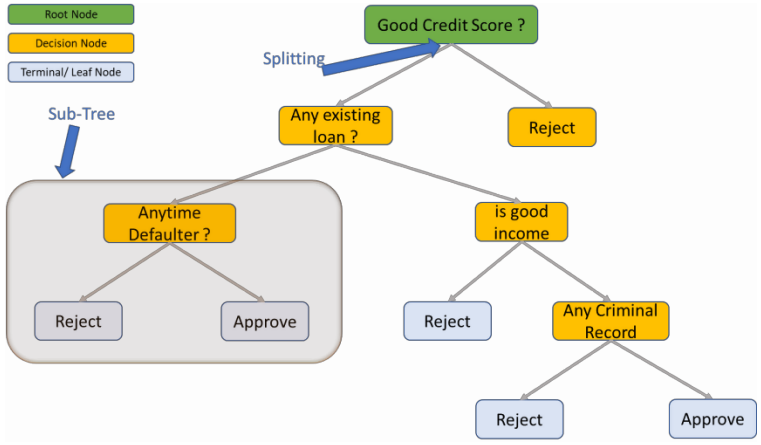
Consider a scenario where we want to decide whether a loan should be approved for an applicant or not. To decide, we will ask the applicant a series of questions. We might start off with whether the applicant has any other existing loans. If the answer is yes, then the next question might be whether he is a defaulter or not. With this kind of series of questions, we can narrow down the search and make a robust decision.



We built the model above by hand to make a decision about whether a loan application should be approved or not. Alternatively, a machine learning model could perform supervised learning using the dataset to arrive at a decision. The decision tree model in machine learning can learn these decision conditions (test) from the dataset quickly and build the model.

Decision tree models work best if the training dataset is in binary format; however, this is not a limitation. For continuous features, the decision condition can be applied in the form of greater than or less than a certain threshold, e.g., $X_1 > 0.7$.

In a decision tree, the topmost node is called the root node, the nodes where the decision tree ends are called leaf nodes/terminal nodes, and all other remaining nodes are called decision nodes. A decision tree can have a sub-branch or sub-tree as well, and the process of creating a sub-branch is called splitting. Have a look at the figure below to visualize these notations.



The root node has the full dataset, and at each decision node, the test is conducted to split the dataset. Decision nodes are executed to split the data until it reaches the leaf node. A leaf node contains a single target value (a single class or a single regression value). A leaf node that contains only one target value is considered pure.

A prediction on a new data point is made by traversing through the decision tree and checking at each decision node whether the condition is met or not.

Decision trees are prone to overfitting if the parameters are set to favor all pure leaf nodes, which means that all data points in the training dataset are correctly classified. To reduce overfitting, the following strategies are commonly followed:

1. Pre-pruning – Stopping the creation of the tree. This is achieved by limiting the maximum depth of the tree, limiting the maximum number of leaf nodes, or defining the minimum number of points required to split it further.

2. Post-pruning – Trimming the nodes that contain less information.

It is important to carefully consider which feature will be used to split each node because decision trees with a different split node may result in different predictions and accuracy. We can utilize a statistical method to identify the feature that should be selected as the root node. The feature that has the most information gain should be selected as the root node. Information gain measures how well a certain feature distinguishes among different target classifications. Information gain is measured in terms of the expected

reduction in the entropy or impurity of the data. The entropy of a set of probabilities is:

$$H(p) = -\sum_{i} p_i log_2(p_i)$$

where *p* is the probability of outcome event *i*.

To understand how entropy and information gain is calculated, let us look at the following example:

> A training dataset has 500 observations. Of these 500 observations, 300 are of positive class, and the remaining 200 are of negative class.
>
> Positive class ratio = 300/500 = 0.6
>
> Negative class ratio = 200/500 = 0.4
>
> Entropy of the target variable E_T = - [0.6 * log2 (0.6) + 0.4 * log2 (0.4)] = 0.9702
>
> A feature X1 in the dataset is split as X1 > 347 (120 positive and 80 negative), X1 <= 347 (240 positive and 60 negative)
>
> Entropy (X1 > 347) = E1= -1* [120/200 log_2(120/200) + 80/200 log_2(80/200)]
>
> Entropy (X1 <= 347) = E2 = -1* [240/300 log_2(240/300) + 60/300 log_2(60/300)]
>
> Entropy of X1 = E_{X1} = 200/500 * E1 + 300/500 * E2
>
> Information gain for feature X1 = E_T- E_{X1}

Whichever feature in the node that has maximum information gain will be selected for the split.

If we have a set of binary responses from some variables, all of which are positive/true/1, then knowing the values of the variables does not hold any predictive value for us since all the outcomes are positive. Hence, the entropy is zero ($\log_2(1) = 0$). If half of the records are of positive class and another half of the records are of negative class, then the entropy is 1 (highest).

The entropy calculation tells us how much additional information we would obtain with knowledge of the variable.

Misclassification rate, Gini index, and iterative dichotomiser 3 (ID3) are other popular methods of calculating the information gain of a feature.

Decision trees are helpful for identifying what features play an important role as well as understanding why a class or value is predicted by the decision tree. Hence, decision trees play an important role in model explanation.

Pre-pruning parameters are the main hyperparameters of a decision tree model, i.e., the maximum depth of the tree, maximum number of leaf nodes, and minimum number of data points required to split the node further. A combination of these hyperparameters can be used to build the decision tree, which is generalized over the dataset and provides good accuracy.

Benefits of Decision Trees

1. Decision trees are helpful when a model explanation is required. Since it is built on various decision

conditions, we can identify what conditions it has tested and what conditions have passed to reach to the decision.

2. Since each feature is processed separately, normalization or standardization of features is not needed.

Disadvantages of Decision Trees

1. Decision trees are prone to overfitting and provide poor generalization performance.
2. They have low prediction accuracy.
3. A decision tree with many class tables can become very complex.

Ensemble

Pre-pruning and post-pruning strategies are implemented to reduce overfitting, but still, decision tree models tend to overfit. To overcome this, an advance modeling technique called ensemble is used. The idea of ensemble is to build many trees—all of which predict well and overfit in their own way—and average the results to reduce overfitting.

Ensemble means assembling many machine learning models (also known as base estimators) to create a more powerful and robust model. The most popular ensemble techniques are discussed below.

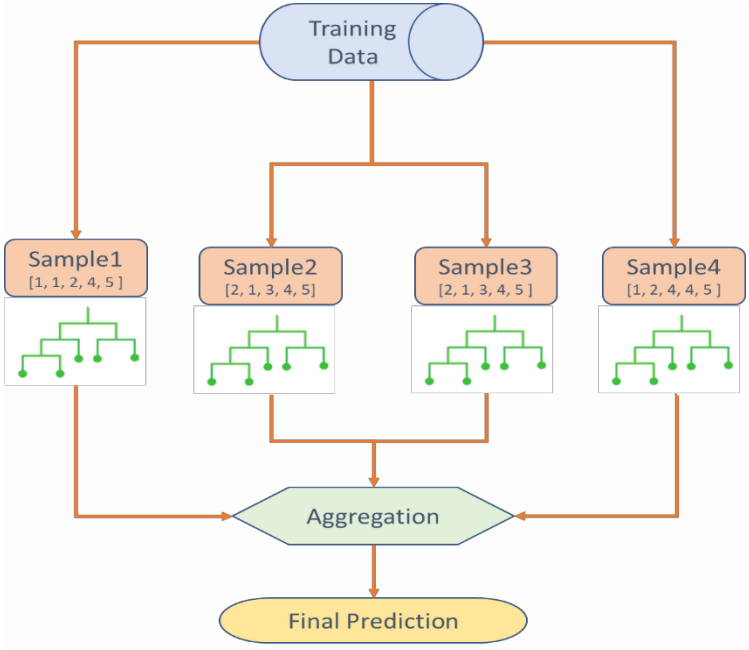
Bagging

Bagging combines bootstrapping and aggregation to form an ensemble model. Multiple bootstrapped samples are drawn (randomly with replacement) from the training dataset and, for each of these samples, a decision tree (base estimator) is created. Finally, results from these base estimators are aggregated to achieve an efficient predictor. Typically, the combined estimator is better than any of the single base estimators.[ix]

Samples from the dataset are drawn in a bootstrap manner, e.g., a sample of 10 observations is drawn from a training dataset of 100 observations. These observations are then returned to the training dataset before another sample is drawn. So, the next sample of 10 observations is drawn from a training dataset of 100 observations. In simple terms, at any point, all of the training dataset observations will be available for a sample to be drawn.

To the aggregate output of base learners, voting is used for classification use cases and averaging is used for regression use cases.

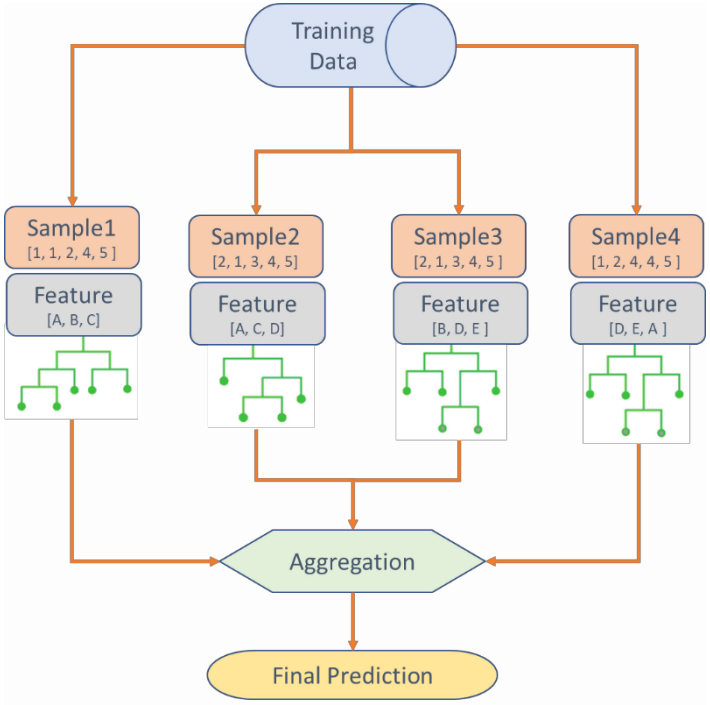
This is useful because models are created on various samples drawn from a single population. Bagging can reduce variance with little or no bias.



Random Forest

Random forest models use the bagging concept with slight modification. During bagging, all features of the training dataset are used on bootstrapped data (sample data) to create the base estimators. Since these sample datasets are quite similar, each base estimator usually breaks at the same feature. This results in quite similar base estimators. This means that weak features will not be incorporated. To avoid this, random forest is used widely. It creates the bootstrapped samples like bagging does but along with this, a subset of features is selected randomly for each node in the decision tree. This selection of a subset of features is repeated separately in each node so that each node in a tree can make a decision using a different subset of features. This process of randomly selecting sample data and

a number of features for a split at each node ensures that all decision trees in the random forest are different.



Similar to the decision tree, the random forest also provides feature importance, which is computed by aggregating feature importance over the trees in the forest. Typically, the feature importance provided by random forest is more reliable than the one provided by a single tree.

The maximum number of features to split at each node determines how random each tree is and a smaller value reduces overfitting. As a rule of thumb, this parameter can be set to the square root of a number of features for classification and $\log_2$(number of features) for regression use cases.

Criterion (Gini, entropy), number of decision trees (n_estimators), maximum number of features (max_features), maximum depth of the decision tree (max_depth), minimum number of samples required at each leaf node (min_samples_leaf), minimum number of samples required to split a node (min_samples_split), and maximum number of leaf nodes in each decision tree (max_leaf_nodes) are all hyperparameters of ensemble models. This hyperparameter can be tuned to get a generalized and accurate machine learning model.

Benefits of Random Forest

1. Ensemble models are very powerful and often work without parameter tuning.
2. Feature scaling is not required.

Disadvantages of Random Forest

1. It is difficult to understand thousands of trees and explain the decision-making process.
2. Ensemble methods need more computing resources and take more time to learn from data.

Boosting

Unlike in bagging, where base estimators are executed parallelly, in boosting, base estimators are executed sequentially, and each subsequent estimator focuses on the weakness of the previous estimator. Boosting incrementally builds an ensemble by training each model with the same dataset but where the model coefficients of estimators are

adjusted according to the error of the last prediction. Several weak models team up to produce a powerful ensemble model. The main idea of boosting is to focus on the observations that are hard to predict. Boosting can reduce bias without incurring higher variance.

Popular boosting algorithms are AdaBoost and gradient boosting.

AdaBoost is adaptive boosting, where more attention is given to the records that are not correctly predicted. After each iteration, weights of the wrongly predicted observations are increased so that these records will be picked up more in the next iteration to gain better accuracy.

Gradient boosting is another popular boosting algorithm. It works by sequentially adding the previous predictor's underfitted predictions to the ensemble, ensuring errors made previously are corrected.

Boosting does not introduce randomness to the decision trees; however, it uses strong pre-pruning techniques to build accurate predictors. In most cases, the maximum depth for boosting models is kept to five models. This makes the model faster, and the model consumes less memory.

These models are more sensitive to hyperparameters, but once the hyperparameters are tuned properly, these models provide very good accuracy and generalization.

Number of decision trees (*n_estimators*), maximum depth (*max_depth*), and learning rate (*learning_rate*) are major hyperparameters of ensembled models with boosting. Learning_rate and n_estimators are dependent on each other. A lower learning_rate means more trees are required and a

higher learning_rate means less trees are required to build a model. These hyperparameters should be tuned to get an optimized machine learning model.

Benefits of Boosting

1. Ensemble models are very powerful and widely used.
2. Feature scaling is not required.

Disadvantages of Boosting

1. It is difficult to understand thousands of trees and explain the decision-making process.
2. Careful tuning of the hyperparameters is required.
3. Boosting is not good for high dimensional sparse data.

Support Vector Machines

Support vector machines, commonly known as SVMs, are supervised learning algorithms. They are mostly used for classification tasks. However, SVM algorithms can be used for regression as well. A support vector machine for classification is called a support vector classifier (SVC), and for regression, it is called a support vector regressor (SVR). Support vector machines can be used to build more complex models that are not simply defined by hyperplanes in input space.

Linear Support Vector Machines

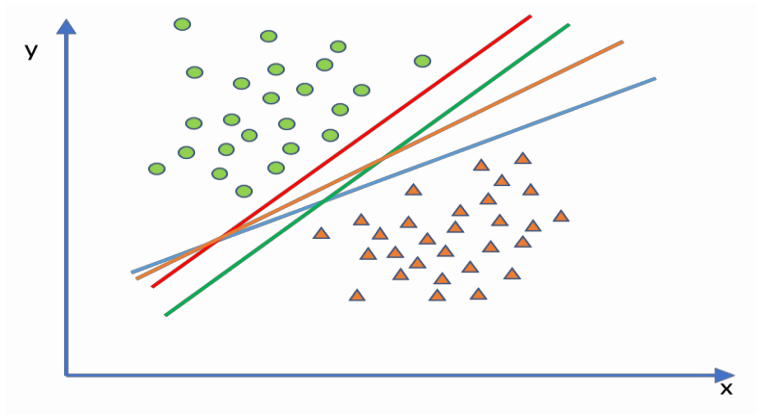
Linear models can perform classification using a line, plane, or hyperplane, but these models do not perform well for non-linear datasets.

In logistic regression, if the data point is closer to the line, then the confidence level of predicting accurate class is low, i.e., if the probability is near 0.5, then the prediction confidence is low. Support vector machine algorithms overcome this situation by maximizing the distance between the classifier (line or hyperplane) and the nearest data points of each class. These selected data points are called support vectors, and the Euclidian distance between the classifier and the closest support vector is called a margin. The result of SVM is the hyperplane for which the margin is highest. A higher margin means there is more confidence in predicting the class accurately.

The number of these support vectors is very small, and, eventually, the hyperplane is identified from these support vectors alone. If any of the support vectors are removed, it will

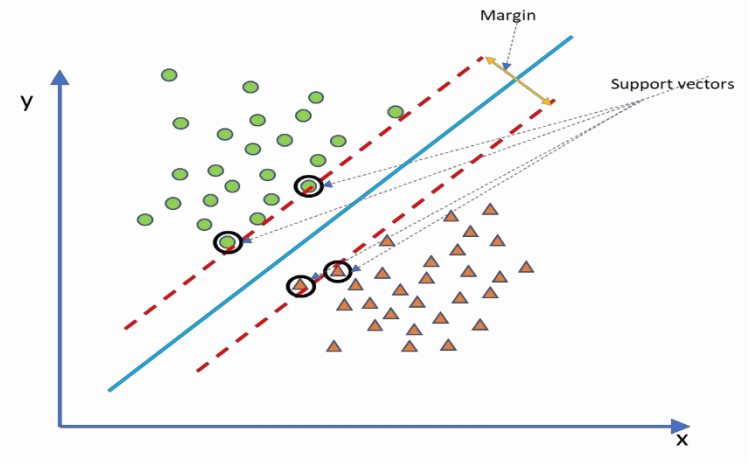
alter the position of the hyperplane. The hyperplane is a linear classifier built by an SVM algorithm. However, an SVM algorithm can be extended to non-linear classification using a kernel trick.

In this algorithm, each data item is plotted as a point in n-dimensional space, where n is the number of features in the dataset, and the value of each feature of the observation is the value of a coordinate. We identify a hyperplane that can distinguish the classes accurately.



For a classification use, we can have multiple hyperplanes that can distinguish the data classes, but which is the best for classifying future data points accurately?

To get the best hyperplane, SVMs need to work on maximizing the distance between the nearest data point and the hyperplane (the margin).



As shown in the figure above, SVMs try to navigate a line that has the highest margin, i.e., the maximum distance between two dotted lines.

A hinge loss function for training classifier is used to maximize the margin.

$$L(x, y, f(x)) = (1 - y * f(x))$$

Here, *L* is the loss function, *y* is an actual class, and *f(x)* is the predicted class. If the result of this loss function is less than 0, then the result is set to 0. It can be rearranged as:

$$L(x, y, f(x)) = \begin{cases} 0, & y * f(x) >= 1 \\ (1 - y * f(x)) & \text{else} \end{cases}$$

f(x) is a function of features and their weights. It can be denoted as:

$$f(x) = h\langle x_i, w \rangle$$

Objective function (loss function with regularization):

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \, h\langle x_i, w \rangle)$$

Objective function has two terms. The first term is a regularization term and the second is a loss term. If regularization is too high, then the model will become overfit, and if the regularization is too low then the model will become underfit. Hence, we must find an optimal value of regularization so that the model can predict accurately as well as generalize at the same time.

We will use a gradient descent technique to minimize the objective function:

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} (1 - y_i \, h\langle x_i, w \rangle) = \begin{cases} 0, & \text{if } y_i \, h\langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

This means if we have misclassified a sample, we update the weight vector *W* using the gradient of both terms; otherwise, if classified correctly, we just update the weight vector *W* by the gradient of the regularization term.

Including the learning rate, the weight vector can be updated as follows:

$$w = w + \eta(y_i x_i - 2\lambda w)$$

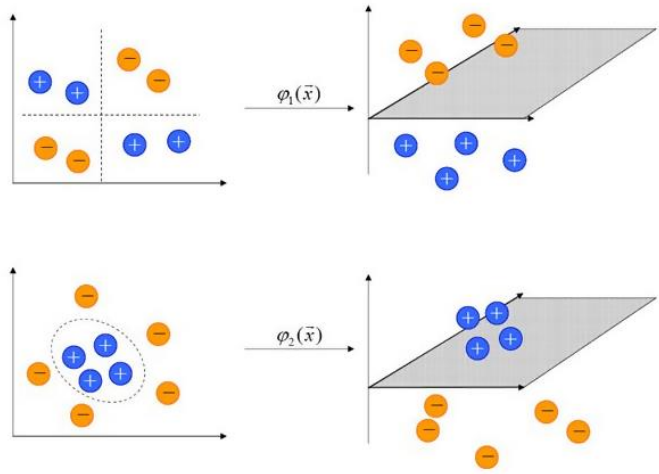
Support vector machines are great for relatively small datasets with fewer outliers. Decision trees and random forests can produce robust classifiers, but they require more data.

Typically, only support vectors matter for defining the hyperplane. To make a prediction for a new point, the distance to support vectors is measured. A decision is made based on the distance to the support vectors, and the importance of the support vectors that was learned during training.

Non-Linear Support Vector Machines

Up to this point, we have discussed linear support vector machines where classes are separated by a hyperplane in a multi-dimensional coordinate system. These SVM algorithms can be extended to non-linear datasets using a kernel trick.

The dataset that is not linearly separable is transformed into a higher dimensional dataset where the classes become linearly separable.



Original feature space High dimensional feature space

There are two kinds of kernels: the polynomial kernel, which computes all possible polynomials up to a certain degree of the original features (like feature1 **2 * feature2 ** 5), and the radial basis function (rbf) kernel, also known as Gaussian kernel, which is widely used for non-linear datasets. The Gaussian kernel uses polynomials of all degrees, but the importance of the features decreases for higher degrees.

Important hyperparameters in SVMs are learning rate, regularization parameter, choice of kernel, and kernel parameters.

Support vector machine models work only for two class classifications. However, in a multi-class scenario, it can create as many models as classes to compare one with the rest, e.g., for Class 1, a model will be built to predict either Class 1 or not Class 1. Similarly, for Class 2, a model will be built to predict either Class 2 or not Class 2. For n classes, n models will be created.

Benefits of Support Vector Machines

1. Support vector machines can create very complex decision boundaries for low dimensional datasets as well as for high dimensional datasets.
2. They are effective for cases where the number of samples is less than the number of features.
3. Support vector machines maximize the distance between two classes and thus provide more confidence level support to predicted values.

Disadvantages of Support Vector Machines

1. Support vector machines do not scale for a huge number of samples. They need a lot of memory and computation for large datasets.
2. They expect all features to be on a similar scale.
3. Careful tuning of the hyperparameters is required.
4. Support vector machine models are difficult to explain.

k-Nearest Neighbors

The k-nearest neighbors (kNN) algorithm belongs to the family of instance-based, competitive learning and lazy learning algorithms.[x]

It is an instance-based algorithm because the model saves all training data points as part of the machine learning model.

It is a competitive learning algorithm because it internally uses competition between model elements (data instances) to make a predictive decision.

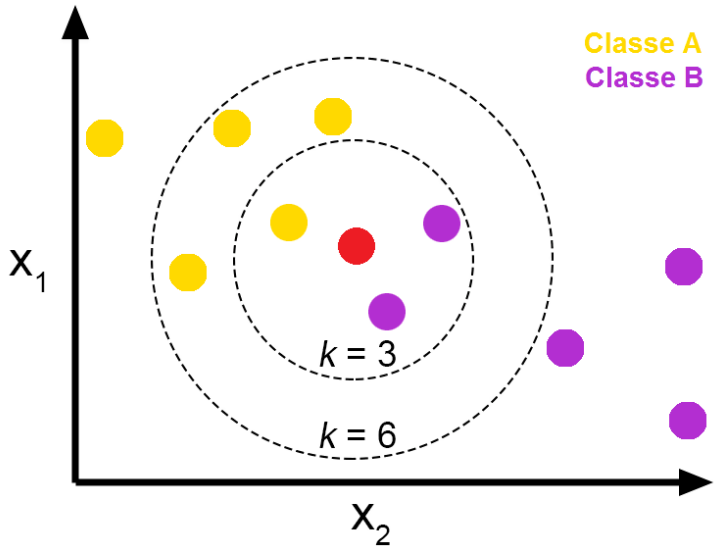
It is a lazy learning algorithm because the calculation is delayed until a prediction is required. This is called a localized model because only the data points that are near new data points are used for model calculation and for predicting classes for new data points.

The model for kNN is the entire training dataset. When a prediction is required for an unseen data point, the kNN algorithm will search through the training dataset for the k-most similar neighbor (k-msn). The prediction attribute of the most similar data points is summarized and returned as the prediction for the unseen instance.

In this model, k is the number of neighbors we want to check to classify a new data point. If $k > 1$, the model uses voting to classify the new data point. In short, the class that is in the majority in k-msn is assigned to the new data point.

For example, if the value of k is set to 3, the model will check the three nearest data points to classify the new data point. The default value of k is 1, which means that the vanilla kNN model

classifies the new data point according to the class of the nearest neighbor.



In the figure above, the yellow and purple circles represent the data points from the training dataset, and we want to predict the class for the red data point. If the value of k is set to 3, then the model will check the three nearest data points (inner circle) and then classify the red data point. If the value of k is set to 6, then the model will check for the nearest six data points (outer circle) and then classify the red data point.

In the figure above, the red point will be classified as follows:

If k = 3, Class B (2 votes for Class B, and 1 vote for Class A)

If k = 6, Class A (2 votes for Class B, and 4 votes for Class A)

The above explanation is for two class datasets; however, the same concept can extend to a multi-class dataset as well. In a multi-class dataset, we count how many data points belong to

each class, and the class that is in the majority is predicted for the new data point.

For continuous features, Euclidian distance is calculated, and for categorical features, Hamming distance is calculated.

Important hyperparameters are:

- *n_neighbors:* It holds the value of k, we need to pass, and it must be an integer. The default value of this parameter is 5.
- *Weights*: It holds a string value, i.e., name of the weight function. The Weight function is used in prediction. It can hold values like "uniform" or "distance" or any user-defined function.[xi] The default value of weights is "uniform."
    - *Uniform* weight is used when all points in the neighborhood are weighted equally.
    - *Distance* weight is used for giving closer neighbors a higher weight and further neighbors less weight, i.e., weight points by the inverse of their distance.
    - *User defined function* can used when we want to produce custom weight values. It accepts distance values and returns an array of weights.[xii]
- *Algorithm:* It specifies the algorithm, which should be used to compute the nearest neighbors. It can hold values like "auto," "ball_tree," "kd_tree," and "brute." It is an optional parameter.

- *Ball tree , kd tree* are used to implement ball tree algorithms. These are special kinds of data structures for space partitioning.
- *Brute* is used to implement brute-force search algorithms.
- *Auto* is used to give control to the system. It automatically decides the best algorithm according to values of training.

Benefits of k-Nearest Neighbor

1. Simple to implement.
2. Flexible to feature/distance choices.
3. Naturally handles multi-class use cases.
4. Can do well in practice with enough representative data.

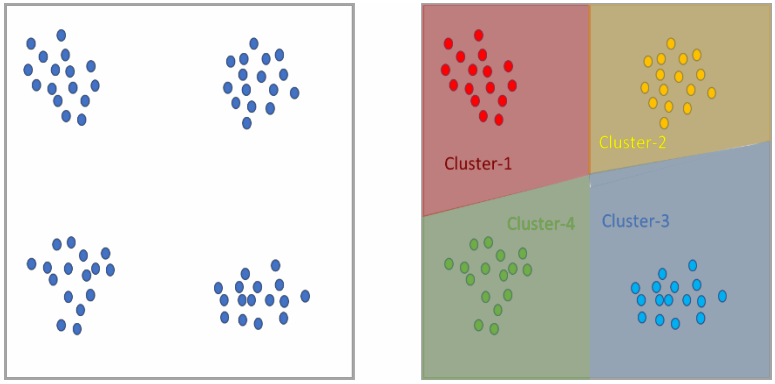
Disadvantages of k-Nearest Neighbor

1. Storage of data and that is is space-consuming
2. For each prediction, the calculation is done separately.
3. Large search problems to find nearest neighbors will take a long time

Clustering and k-Means

In most of the cases, labeling data is a manual task and it cost huge money to organizations. Hence it is imperative to have algorithms which can work on unlabeled data. Clustering and K-means are the most commonly used machine learning algorithm to analyze unlabeled data and get useful insights from it.

Clustering can be correlated to grouping, means the data points which are similar in some way and different from other data points are grouped together. Each group of the data points is known as a cluster. Clusters can be visualized as below.



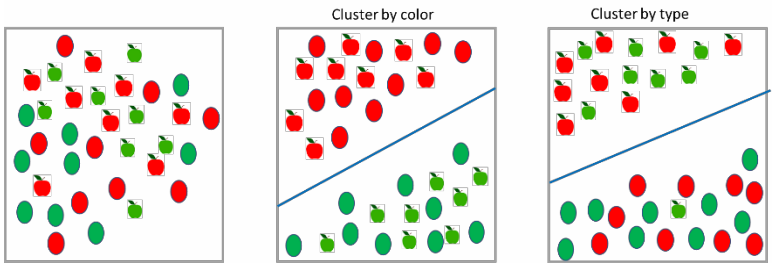
In the above image, data points are divided into four clusters based on the geometrical distance between two data points. The data points which are close to each other are assigned to one cluster. This approach of creating clusters is known as distance-based clustering.

Another approach of creating clusters is known as conceptual clustering. In this approach, clusters are created based on conceptual similarity. e.g. in a dataset of oranges and apples,

two clusters will be created. One for all apples and another one for all oranges. This approach uses properties of data points to distinguish one data point to another and to group data points with similar properties together.

Clustering is a very subjective area of discussion. It is difficult to determine how one clustering is better than the other one. Machine learning algorithm learns the insights from the data and applies these insights to create clusters. Since we don't know those insights, it is hard to say the clustering done by the machine learning model is best or not.

The user must provide an appropriate criterion to get suitable clusters as an output from the machine learning model. E.g. consider a dataset contains green color balls, red color balls, green color apples, and red color apples. Now if we provide color as the clustering criteria to the machine learning model then it will create two clusters, one of the green color balls and green color apples and the second one of red color balls and red color apples. However, if the criteria provided is 'edible' then the machine learning model will create two clusters, one of all apples and another one of all balls.



Clustering is also used for finding outliers (unusual data points) and finding homogeneous groups.

Clustering algorithm can be applied in many industries e.g. marketing (finding group of customers with similar behavior), biology (plants classification), insurance (fraud detection), city planning, earthquake studies, document classification, etc.

k-Means Clustering

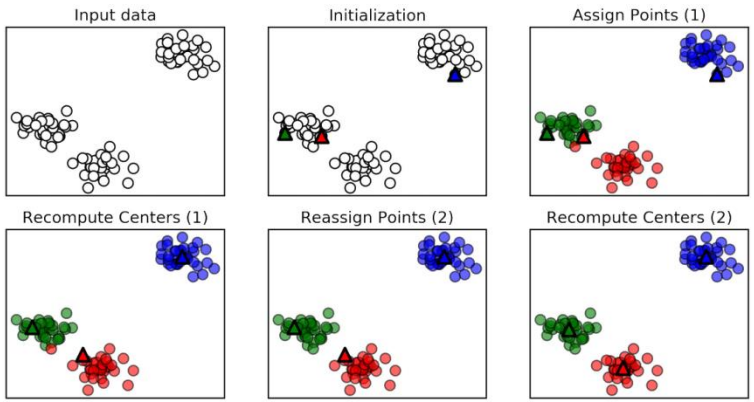
k-means clustering is one of the most useful clustering algorithms. It tries to find cluster centers that represent certain regions of the data.

k-means clustering is a two-step process:

1. Assign each data point to the closest data center.
2. Set the cluster center as the mean of the data points assigned to the cluster.

The algorithm keeps on executing the above steps until the assignment of data points to clusters no longer changes.

Once we decide how many clusters are required, the number of clusters is passed to the algorithm. Let us say for a given dataset we want three clusters.



- In the initialization step, the algorithm randomly selects three cluster centers.
- Each data point is assigned to the cluster center it is closest to.
- We then update the cluster center with the mean of the cluster.
- The previous two steps are repeated until the assignment of data points to clusters no longer changes.

Clustering is similar to classification; the only difference is that classification use cases have labeled data (class defined) and clustering creates various clusters from the dataset.

Assumptions in k-means algorithms:

- All directions are equally important for each cluster.
- All clusters have the same diameter. The boundary between clusters is always drawn exactly in the middle of the cluster centers.

Benefits of a k-Means Algorithm

1. Easy to implement.
2. Efficient for large datasets.
3. Terminates at local optimum.

Disadvantages of a k-Means Algorithm

1. Dealing with a large number of dimensions and a large number of data items can be problematic because of time complexity.
2. The effectiveness of the method depends on the definition of "distance" (for distance-based clustering).
3. If an obvious distance measure does not exist we must "define" it, which is not always easy, especially in multi-dimensional spaces.
4. The result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.
5. The number of clusters must be known in advance.
6. It is unable to handle noisy data and outliers.
7. Not suitable for clusters with non-convex shapes.

Chapter 6: Model Performance

In previous chapters, we discussed how machine learning algorithms work and how these algorithms can be used to build models. We can apply (manually or automatically) various models on a given dataset before finalizing a model. But how do we evaluate the performance of a model to determine that one model is performing better than the other?

In this chapter, we will discuss how to evaluate the performance of a model and the matrices that are used for performance evaluation of a model.

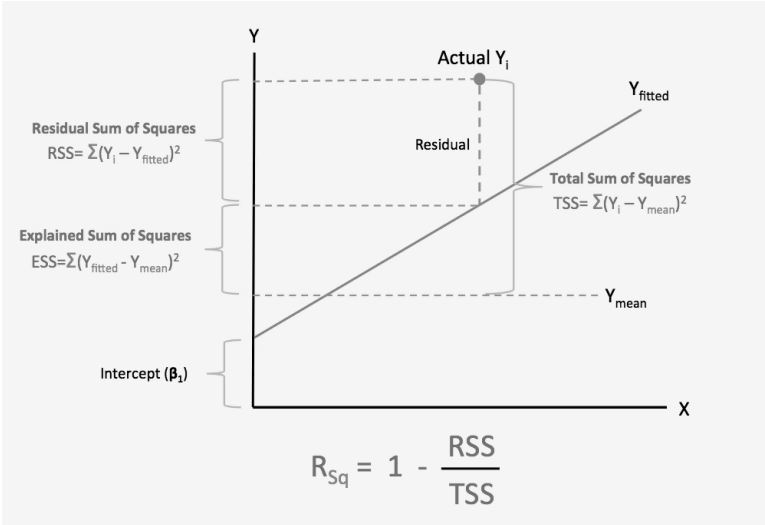
R-squared

The most common way to evaluate the performance of a linear model is by the R-squared (R^2) value. It is a statistical measure of determining how close the data is to the fitted regression line. It is a proportion of explained variance to total variance.

$$R^2 = \frac{\text{Explained Variance}}{\text{Total Variance}}$$

Here, the explained variance is the variance in the observed data that is explained by the model, and total variance is the variance present in the observed data.

Here is how it is represented graphically:



Typically, the R- squared value lies between 0 and 1:

- 0 means the model does not explain any variability of the data around its mean.
- 1 means the model explains all the variability of the data around its mean.

The higher the value of R-squared, the better the model fits the data and the better the model explains the variance of the observed data around its mean. However, it has a few limitations:

- R-squared cannot determine whether the coefficient estimates and predictions are biased.
- R-squared is influenced by a number of features/predictors. Adding more features increases the R-squared value. If the number of features is too high, then the model starts predicting random noise in the data and tends to overfit.

- It is difficult to tell whether an increase in the R-squared value is a result of better model performance or more features in the model.

Adjusted R-squared

To overcome the influenced behavior of R-squared on the number of features, Adjusted R-squared is used. Adjusted R-squared is a modified version of R-squared that has been adjusted for the number of features in the model.

Adjusted R-squared penalizes the R-squared if the choice of the feature (newly added to model) is not good.

$$R^2_{adjusted} = 1 - \frac{(1- R^2)(n-1)}{(n-k-1)}$$

Where
n= no of observation
K = no of features in the model
R^2 = R-squared

Hence, for linear regression scenarios, adjusted R-squared should be used to evaluate the model performance.

Confusion Matrix

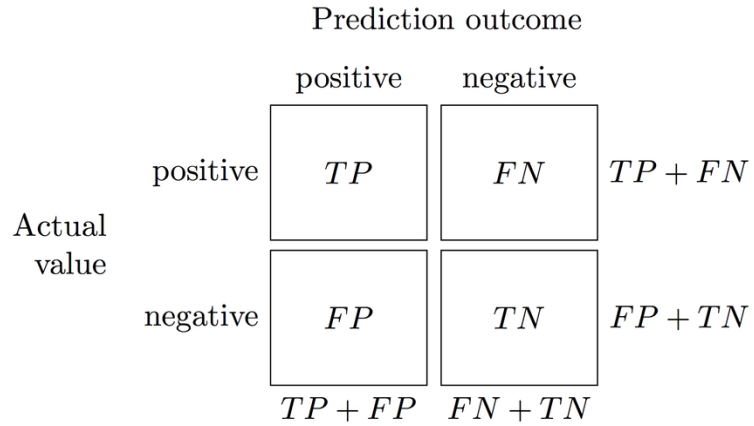
In classification scenarios, having good R-squared or adjusted R-squared values is not important until both the classes are unidentified with similar accuracy. This becomes a big issue when the data is not equally distributed for the target classes. For example, consider a dataset with 98% observation of Class

A and 2% observation of Class B. The model can easily get 98% training accuracy by simply predicting that every sample belongs to Class A, but the prediction accuracy for Class B is very poor.

To overcome this situation, model performance should be evaluated for each class and then aggregated to get the overall performance of the model.

Confusion matrix provides the right tools/functions to get the individual and overall performance of a classification model.

The confusion matrix is depicted as:



For a two-class (positive and negative classes) classifier, prediction and actual values are represented as shown in the above figure. The total number of positive classes is P, and total number of negative classes is N.

TP (True Positive) → Actual class is positive, and predicted class is also positive (top left corner in the above figure). This states how many positive classes correctly predicted. This is also known as the power of the model.

FN (False Negative) → Actual class is positive and predicted class is negative (top right corner in the above figure). This states how many positive classes are predicted as negative. This is also known as a Type II error (miss).

FP (False Positive) → Actual class is negative, and predicted class is positive (bottom left corner in the above figure). This states how many negative classes are predicted as positive. This is also known as a Type I error (False alarm).

TN (True Negative) → Actual class is negative and predicted class is also negative (bottom right corner in the above figure). This states how many negative classes are correctly predicted.

Correctly predicted classes → TP + TN

Incorrectly predicted classes → FP + FN

Actual positive classes → P = TP + FN

Actual negative classes → N = FP + TN

Accuracy – This indicates the proportion of records that are correctly classified. It is calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

The drawback with accuracy is that it does not indicates the accuracy of each class. In a highly biased dataset, accuracy cannot state whether both classes are correctly identified with the same accuracy or not, e.g.:

> Total no. of observations → P = 98, N = 2 (TP + TN + FP + FN = 100)

Actual positive records correctly identified→ TP = 95

Actual negative records correctly identified→ TN = 0

Accuracy → (95 + 0)/100 = 0.95

In the above example, the accuracy of the model is 95% but the accuracy of identifying a negative class is 0.

Recall or true positive rate (TPR) – This is a measure of how many actual positive classes are correctly identified. It is defined as the ratio of the number of correct positive class predictions to the number of actual positive classes. High recall means a high rate of correctly predicting positive class. This metric is important when we want to avoid a false negative. It is also known as sensitivity and hit rate.

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$

Precision or positive predictive value (PPV) – This is a measure of how many positive predicted classes are actually positive. It is defined as the ratio of the number of correct positive class predictions to the number of positive class predictions. High precision means incorrectly predicted positive classes (FP) are low. This metric is important when we want to avoid a false positive.

$$Precision = \frac{TP}{TP + FP}$$

f1-score – Precision and recall are important measures but looking at only one of them does not provide a clear picture of model performance. The most common way of summarizing accuracy and recall is the harmonic mean of these two measures. It is a better measure than accuracy because it takes precision and recall into account. This is also known as an *f1*-score. It is calculated as:

$$f1\text{-score} = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

For example:

Total no. of observations → P = 98, N = 2 (TP + TN + FP + FN = 100)

Actual positive records correctly identified→ TP = 95

Actual negative records correctly identified→ TN = 0

Actual positive records identified as negative→ FN = 3

Actual negative records identified as positive→ FP = 2

Accuracy → TP/(P + N) = (95 + 0)/100 = 0.95

f1-score → 2TP/(2TP + FP + FN) = (2*95)/(2*95 + 3 + 2) = 0.97

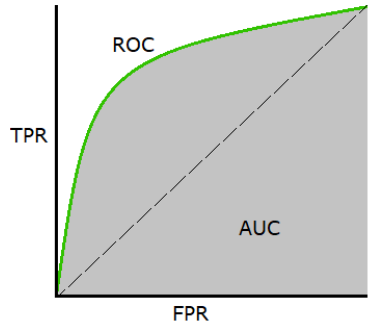
Receiver Operating Characteristic Curve and Area Under the Curve

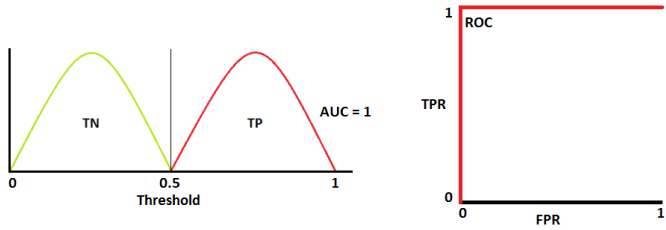
In classification, the default probability of classifying into positive and negative classes is 0.5. This is also known as a threshold, e.g., $p > 0.5 \rightarrow$ Class 1, $p < 0.5 \rightarrow$ Class 0. The threshold value of 0.5 does not hold good for all scenarios. We can change this threshold value to get the desired value of recall and precision. But while developing a new model, desired values of recall and precision are not known. To overcome this and to better understand the behavior of the classifier at different thresholds, the receiver operating characteristic (ROC) curve is used. This is a plot between the false positive rate (FPR) and TPR for a given threshold.

$$FPR = \frac{FP}{FP + TN} \qquad TPR = \frac{TP}{TP + FN}$$

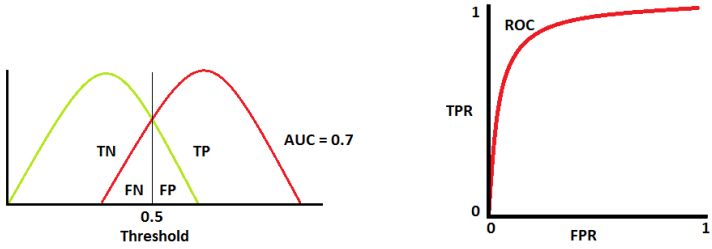
The area under the curve (AUC) of the ROC curve represents the degree of measure of separability, i.e., how much the model is able to distinguish between classes. The higher the AUC, the better the model is at predicting positive class as positive and negative class as negative.

AUC = 1 means the model can clearly distinguish between the two classes.

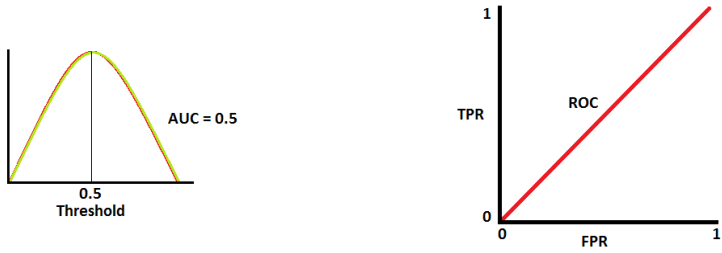




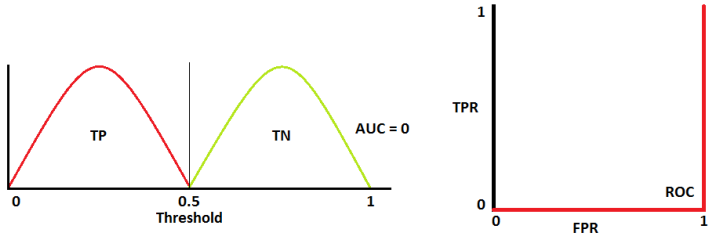
AUC = 0.7 means there is a 70% chance that a positive class will be identified as positive and a negative class will be identified as negative.



AUC = 0.5 means the model cannot clearly distinguish between two classes.



AUC = 0 means the model is identifying positive classes as negative classes and vice versa.

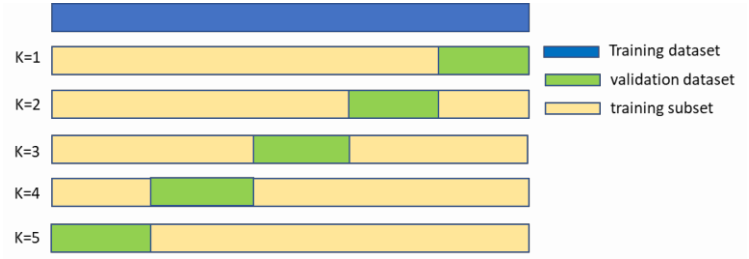


Cross Validation

Cross validation is a tool that utilizes the training dataset in a better way to reduce overfitting and underfitting. It is a model validation technique for assessing how the results of statistical analysis will generalize to an independent dataset.

The purpose of using cross validation is to increase confidence in the model trained by the training dataset. Without cross validation, our model may perform well on the training dataset, but the performance decreases when applied to the testing dataset. The testing dataset is precious and should be only used once, so the solution is to separate one small part of the training dataset as a test of the trained model, which is the validation dataset.

K-fold cross validation – This involves splitting the training dataset into k subsets of data (also known as folds). The machine learning model is trained on k-1 subsets and then evaluated on the subset that was not used for training. This process is repeated k times, with a different subset reserved for evaluation (and excluded from training) each time. Once the model has been executed for all training subsets, the average of error of each run is calculated and represented as the cross-validation error.

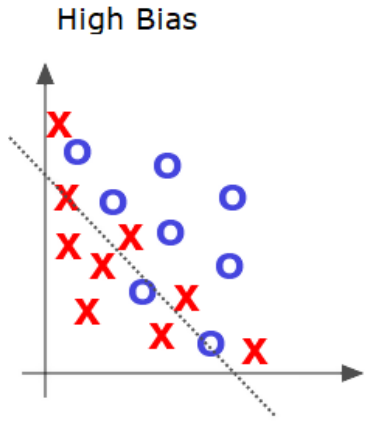


Leave-one-out cross validation – This is another technique used for cross validation. It is a logical extreme of k-fold cross validation where k = n (number of observations). For each run, only one observation is left with a validation dataset. This approach leads to higher variation in testing model effectiveness because testing is done against one observation only. Hence, the estimation is highly influenced by the validation observation. If the validation observation is an outlier, it can lead to higher variation.

Bias

The EliteDataScience defines bias as: "when an algorithm has limited flexibility to learn the true signal from the dataset."[xiii]

Bias is an algorithm's tendency to consistently learn the wrong thing by not taking into account all available information in the dataset. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting). High bias makes the model too generalized, which means predictions will not be accurate. For example, if someone is biased towards a certain brand, then they are more likely to make wrong assumptions about the brand's products.

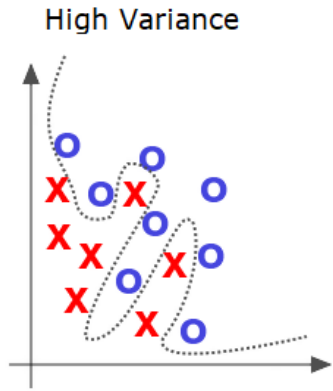


High bias can be reduced by adding more features, adding more polynomial features, or by decreasing regularization.

Variance

According to the EliteDataScience, variance can be explained as follows: "an algorithm's sensitivity to specific sets of the training set occurs when an algorithm has limited flexibility to learn the true signal from the dataset."[xiv]

Variance is the algorithm's tendency to learn random things irrespective of the real signal by fitting highly flexible models that follow the noise in the data too closely. High variance causes an algorithm to model the random noise in the training dataset, rather than the intended output (overfitting). High variance makes the model learn from noise, which means predictions will not be accurate.

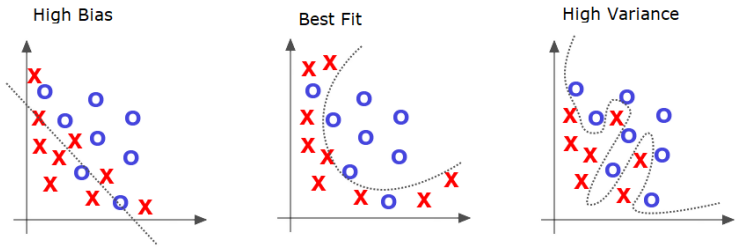


High variance can be reduced by collecting more training examples, using less features, or by increasing regularization.

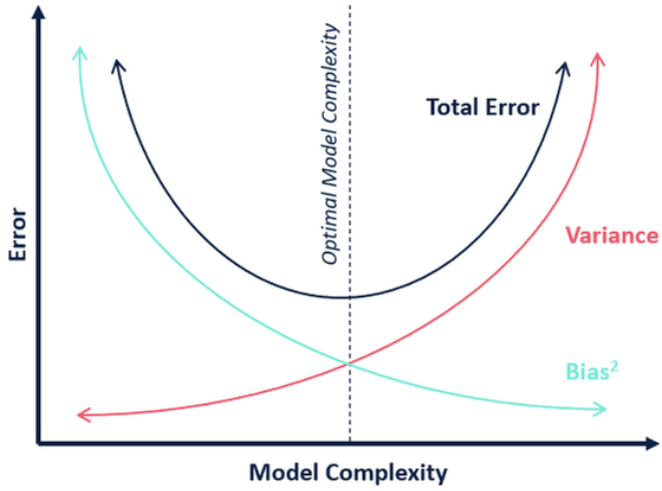
Bias–Variance Trade-off

High bias tends to highly generalize the model, and high variance tends to highly overfit the model. We need an optimal model that is neither highly generalized nor highly overfit. To find this optimal model, trade-off needs to occur between bias and variance.

With a bias–variance trade-off, we want to get the best model that is neither overfitted nor underfitted. It means the model should have low bias and low variance.



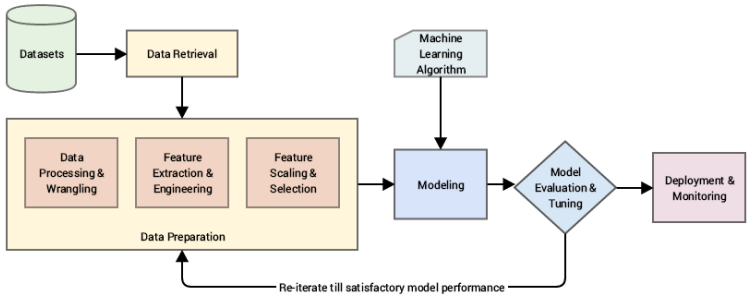
If the model is too simple, it results in underfit; if too complex, it results in overfitting. This means that as the model's complexity increases, bias decreases and variance increases.



As shown in the figure above, we should create an optimal model that is neither too complex nor too simple. This optimal model will have low variance and low bias.

Chapter 7: Best Practices

Many times, the dataset that we get for machine learning problems is not as clean as we expect, and the same data cannot be used to create a machine learning model. We might have to clean the dataset and transform it into a dataset that can be used to build a model. In due course of cleaning and transforming the dataset, we might have to go through multiple processes. This set of processes is part of data preparation. A major part of a machine learning project is devoted to data preparation, since the model performance is governed by the data on which the model is trained. A dataset with good data and good quality features positively influences the model performance. The high level steps involved in a machine learning use case are depicted below.



In this chapter, we will discuss data preprocessing and best practices that we can follow to tackle the datasets and machine learning use cases.

Feature Engineering

Features are the independent variables. We use these independent variables to predict the independent variable, also known as the target. Frequently, the independent variables available in the dataset have hidden information that the machine learning models cannot utilize. To negate this situation in the data preprocessing stage, we apply the domain knowledge and create new informative features out of the existing features available in the dataset. These features should be created carefully; otherwise, the model may overfit. The set of features on which the model is supposed to run should be selected wisely because good features with good quality data can yield a less complex model with better results.

Feature engineering is an art that can make a huge difference between models.

Feature engineering is a recursive process that can be broadly divided into the following steps:

1. Understand dataset
2. Brainstorm features
3. Create new features
4. Validate what impact these features have on the prediction result
5. Restart from Step 1 until the desired accuracy and other metrics are achieved.

For example, consider a model designed to predict salary hikes for employees in an organization. The dataset contains employee ID, geographical location, date of birth, employment start date, career start date, etc. In this dataset, date of birth and career start date might not be useful for a dataset. If we

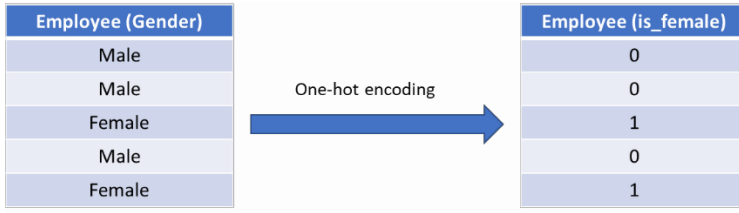
derive two new features, e.g., employee age and years of experience, these two features could play a key role in salary hike predictions. The process of identifying and creating these derived features is called feature engineering.

Feature engineering is an art and it comes with domain knowledge and experience.

One-Hot Encoding

Machine learning models can be trained only on numerical data. These models cannot handle non-numeric features by themselves. So, how do we transform the non-numeric features? e.g., in the employee dataset, gender is maintained as a non-numeric feature, and this feature can have two values: Male or Female. For an organization, an employee's gender is meaningful information. Since it is a non-numeric feature, if we try to train a model on the gender feature then the model will not be able to interpret anything meaningful from it. To make the gender column meaningful to the model, we must transform it into a numeric column.

One-hot encoding is one of the transformation techniques that can be used to transform categorical features into numeric categorical features. If the feature has only two categories, then those two categories can be replaced by 0 and 1. In our employee dataset example, we can replace Male with 0 and Female with 1.



Now, consider a scenario where the categorical column has more than two categories, e.g., in our employee dataset, there is a feature called Last_Rating, which can have the values Needs Improvement (NI), Meeting Expectations (ME), Successful (S), Exceeding Expectations (EE), or Extra Ordinary (EO).

One-hot encoding can be extended to take care of multi-category features. It will create a new feature for each category and, for each observation, the value 1 will be assigned to the newly created feature to which the category belongs to. Other newly created features will be set to 0, e.g., in our employee rating example, five new features will be created (is_NI, is_ME, is_S, is_EE, and is_EO).

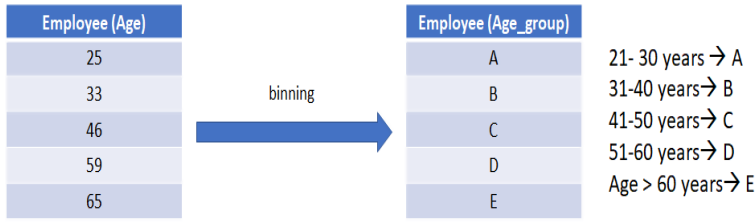
From these five derived columns, one column is obsolete. If four values are known, then the fifth value can be derived. For example, if the is_NI column is removed then the value of the is_NI column can be derived from the other four columns. This means that if all other four columns are 0, then the value is is_NI.

Binning

In a few machine learning scenarios, continuous features cannot be used directly to train a model. These features should be converted into categorical features, and then one-hot encoding should be applied to make these features important for the machine learning model. In our employee dataset example, employee age ranges from 21 to 60 years. These employees can be categorized into four age brackets: 21–30, 31–40, 41–50, and 51–60.

The technique of converting a continuous feature into multiples bins and creating a new feature from it is known as binning or bucketization. It is also known as quantization. Binning transforms a continuous feature into a categorical feature, and categorical feature engineering might need to be performed before using this feature in modeling.

We can create a new age group feature and map each employee with one of these brackets.



For binning, we can use the domain expertise as well as few statistical methods to correctly determine the number of bins/buckets and the boundaries of each bin.

Common methods of binning are:

1. Fixed-width binning – In this technique, width is decided for each bin based on domain knowledge, rules, or constraints.

2. Quantile-based binning – This technique divides the data into q equal partitions. If q = 4 then the parts are quartiles (divide data into four equal partitions).

3. Two-way ANOVA (analysis of variance) test – This is used to find similarity between the various data points of the feature. These similar data points can be grouped together to partition the dataset.

Feature Scaling

Many machine learning algorithms do not perform well when all features in the dataset are not on the same scale, e.g., in our employee dataset example, salary may range from $40,000 to $200,000 and age ranges from 21 to 60. To improve the

performance of these models, all features should be brought to one scale.

Some common normalization (min.–max. scaling) and standardization techniques are used for feature scaling.

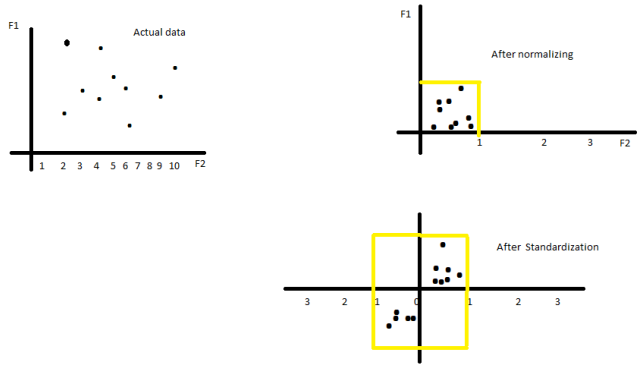
Normalization – In this feature scaling technique, all features are transformed into a 0 to 1 scale. Normalization results in smaller standard deviation, which reduces the effect of outliers. It is achieved by subtracting the minimum value and dividing by the maximum value minus minimum value. It can be represented as follows:

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Standardization – This technique transforms the features so that the mean of the distribution becomes 0 and the standard deviation of the distribution becomes 1. Unlike normalization, standardization does not bound values to a specific range. Standardization is not significantly affected by outliers.

$$X_{changed} = \frac{X - \mu}{\sigma}$$

The figure below shows how the transformed feature will be after implementing normalization and standardization.



Data Imputation Techniques

Datasets for machine learning model use frequently have missing values or outliers. One way to handle this is by discarding those observations, but this may result in a small training dataset; as well, the model will lose out on some valuable information. Another way to handle these missing values and outliers is to use data imputation techniques. There is no good way to handle missing data. However, we will discuss a few common data imputation techniques. Data imputation for categorical and numeric features occurs in different ways.

Numeric features data imputation:

1. Replacing with mean, median, or mode – If a numeric feature has missing values, first of all, check the distribution of the feature's data. If the feature is normally distributed, then missing values can be replaced with the mean value of the feature. If the data distribution of the feature is skewed, this means the feature has outliers and the mean value is influenced by

those outliers. In this case, it is better to replace missing values with the median value of the feature because the median is not significantly influenced by the outliers.

2. Replacing with random sample values – To replace missing values with randomness, we select random observations from the dataset and replace its feature value in the observation that has missing values. To introduce more randomness, we can select a different random observation for each missing value.

3. Replacing with regression – The missing values are obtained by regressing the missing feature on other features. This technique maintains the relationship between all features.

4. Replacing with extrapolation and interpolation – Estimate missing values from other observations of the same feature. These estimations should be made with extra care; otherwise, these will add more assumptions to the dataset, e.g, the age of a person cannot decline; hence, this constraint should be kept in mind before estimating missing values for age.

Categorical features data imputation:

1. Replacing with a new "missing" category, missing – In this technique, a new category "missing" is created for the feature and this value is updated in all observations where a feature's value is not present.

2. Replacing with mode – This technique is good to use when the number of missing values is low. In this

technique, the missing values are replaced by the category that occurs most frequently in the feature.

3. kNN prediction – Apply kNN on other features to predict the feature that has a missing value. An optimal value of k should be decided before running the kNN model.

4. Handling outliers – In this technique, the outliers are replaced with a new category: "rare".

The above-mentioned data imputation techniques are more realistic when data in the feature (for which missing values are to be imputed) can be grouped based on other feature(s). This controls bias in the data imputation.

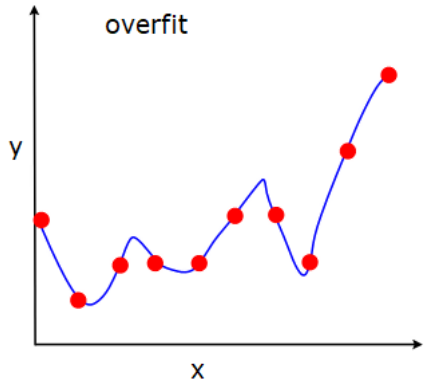
For missing values, it is always better to create a new feature to keep track of which observation has been updated by the data imputation technique. This new feature helps to reduce bias introduced by the data imputation technique.

Overfitting and Underfitting

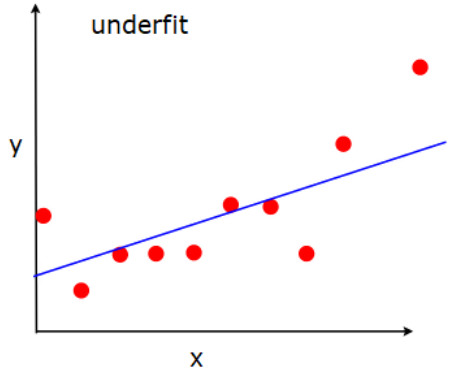
A typical machine learning model learns from the training dataset and applies the learning to the test dataset or unseen data. If the model is able to make correct predictions on unseen data, then the model is able to generalize the learning to unseen data.

When a model learns from the particularities of the training dataset and the noise available in the training dataset, and when it obtains a model that works very well on the training dataset

but is not able to generalize it to test datasets or unseen data, then the model is called an overfitted model.



Contrary to the overfitted model, if the model is too simple and does not learn all the aspects and variations from the training dataset, then the model performs poorly on the training dataset, and it is called an underfitted model.



The goal of modeling is to come up with a model that is not too generalized and not too focused on each individual data point. This model is called a best-fit model. This is basically a trade-off between an underfit and an overfit model.

Regularization

Regularization is a technique used to avoid overfitting of a model by adding a penalty term to the cost function. Regularization helps to generalize a model by bringing down the model coefficients close to 0 so that no feature has more importance than the other features in the model. The most commonly used regularization techniques are ridge (l2) and lasso (l1). Lasso penalizes the l1 norm (sum of absolute values of the model coefficients $\sum_{j=1}^{p}|\beta_j|$); hence, it is called l1 regularization. Ridge penalizes the l2 norm (sum of squared values of the model coefficients $\sum_{j=1}^{p}\beta_j^2$); hence, it is called l2 regularization. Lasso regularization reduces the model coefficient of unimportant features to 0, which means that these features are not used for model building and prediction. Ridge regularization brings the model coefficients close to 0 but does not make them exactly 0. Hence, all features are used in model building and prediction.

Conclusion

Hopefully, this book has demystified the notion of machine learning. But that is just the beginning. Now that you are familiar with the logic behind the different types of learning, understand the role of statistics, and know how to create simple algorithms, it is time to move on. But you will not be alone on that path. We have more books to reinforce your efforts and guide you at every stage.

Machine learning is a crucial development in today's world. The concepts behind it have been around for more than a decade, but the age of machine learning and related models—such as artificial intelligence, data science, and more—is now. The change is just happening and it is fast.

You have made a great decision to start your journey into the world of machine learning with this book. Today, the knowledge and the ability to use machine learning is a competitive advantage. Tomorrow, it will be a mere necessity.

Machine learning techniques have already started to change the world of business, by creating a new value for data. The future will be even more exciting. Very soon, most of the devices and apps that we use daily will be fueled by machine learning algorithms. Many of them already are. Now you have a chance to become a part of this major development. Congrats on your decision and don't forget to check out our other books!

Visit our website at **http://aisciences.net/books/** to find more books.

Next Steps

Thank you for purchasing this book. You now have a baseline understanding of the key concepts of machine learning.

In addition, there is a free bonus chapter available online where you will learn about neural networks and deep learning. You can find this chapter at https://bit.ly/2WoDlik.

If you have any feedback, please let us know by sending an email to review@aisciences.net. This feedback is highly valued, and we look forward to hearing from you.

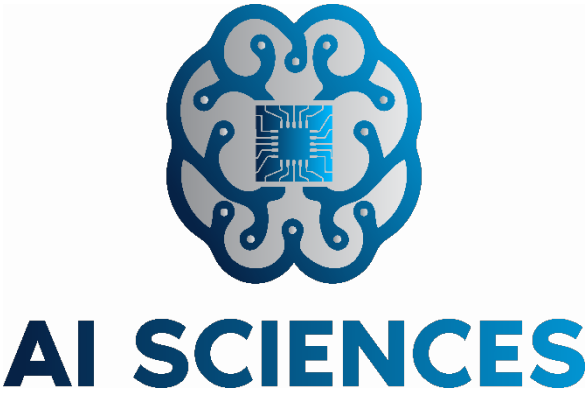
We highly recommend you to visit our website www.aisciences.net and subscribe to our email list. You will receive all of our books for free in an eBook format and you will be informed about all our promotions and offers.

Thank You!

Thank you for buying this book! It is intended to help you learn and master machine learning essentials. If you enjoyed this book and felt that it added value to your life, we ask that you please take the time to review it.

If you noticed any problems with this book, please let us know by sending an email to review@aisciences.net before writing a review online. It will be very helpful for us to improve the quality of our books.

If you want to help us produce more material like this, then please leave an honest review. It really does make a difference.


AI SCIENCES

Sources & References

[i] Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.

[ii] Hybrid Algorithms With Instance-based Classification, https://link.springer.com/content/pdf/10.1007/11564096_19.pdf (accessed March 20, 2019).

[iii] What Is Statistics? | Types Of Statistics | Descriptive ... (n.d.). Retrieved from https://www.youtube.com/watch?v=IngKIlvpg3s

[iv] The example is from: Assimakopoulos, D., Betsos, G., Chalelli, E., Garofalakis, J., Giannoudakis, I., Koskeris, A., & Stamatis, A. (2015). An Integrated Web-Based System for Managing Payrolls of Regionally Spread Governmental Offices. European Conference on E-Government, 489.

[v] Leedy and Ormrod, 2001. / Generating A Research Hypothesis, https://people.uwec.edu/piercech/ResearchMethods/Generating%20a%20research%20hyp (accessed March 20, 2019)

[vi] Distributions Related To The Normal Distribution, http://www.stat.ucla.edu/~nchristo/introeconometrics/introecon_gamma_chi_t_f.pdf (accessed March 20, 2019).

[vii] What Is Likelihood? Definition And Meaning .., http://www.businessdictionary.com/definition/likelihood.html (accessed March 20, 2019).

[viii] Decision Tree Is A Type Of Supervised Learning Algorithm, https://www.greatlearning.in/gl4l-library/decision-tree-for-beginners/ (accessed March 20, 2019).

[ix] Random Forest For Regression[case Study] - 24 Tutorials, https://www.24tutorials.com/machine-learning/random-forest-regression/ (accessed March 20, 2019)

[x] Github - Sammanthp007/stock-price-prediction-using-knn .., https://github.com/sammanthp007/Stock-Price-Prediction-Using-KNN-Algorithm (accessed March 20, 2019).

[xi] Knn Sklearn, K-nearest Neighbor Implementation With Scikit .., https://dataaspirant.com/2016/12/30/k-nearest-neighbor-implementation-scikit-lea (accessed March 20, 2019)

[xii] As above

[xiii] Elite Data Science

[xiv] As above