MASTERING MACHINE LEARNING IN ONE DAY

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 noticed any problem, please let us know by sending us an email at contact@aisciences.net before writing any review online. It will be very helpful for us to improve the quality of our books.

AI SCIENCES

Table of Contents

© Copyright 2019 by AI Sciences
All rights reserved.
First Printing, 2019

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

ISBN-13: 978-1-7335706-8-8
ISBN-10: 1-7335706-8-3

The contents of this book may not be reproduced, duplicated or transmitted without the direct written permission of the author.
Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

I

Legal Notice:

You cannot amend, distribute, sell, use, quote or paraphrase any part or the content within this book without the consent of the author.

Disclaimer Notice:

Please note the information contained within this document is for educational and entertainment purposes only. No warranties of any kind are expressed or implied. Readers acknowledge that the author is not engaging in the rendering of legal, financial, medical or professional advice. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, which are incurred as a result of the use of information contained within this document, including, but not limited to, errors, omissions, or inaccuracies.

- 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 very fast and understand very easily?
- Would you like to practice AI techniques?

If the answers are yes, 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.

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 in big 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 and without a long

reading. Today we also publish more complete books on some topics for a wider audience.

About our Books

Our books have had phenomenal success and they are today among the best sellers on Amazon. Our books have helped many people to progress and especially to understand these techniques, which are sometimes considered to be complicated rightly or wrongly.

The books we produce are short, very pleasant to read. These books focus on the essentials so that beginners can quickly understand and practice effectively. You will never regret having chosen one of our books.

We also offer you completely free books on our website: Visit our site and subscribe for free to our list : www.aisciences.net

By subscribing to our mailing list, we also offer you all our new books for free and continuously.

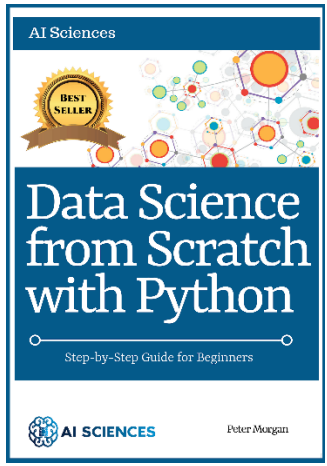
To Contact Us:

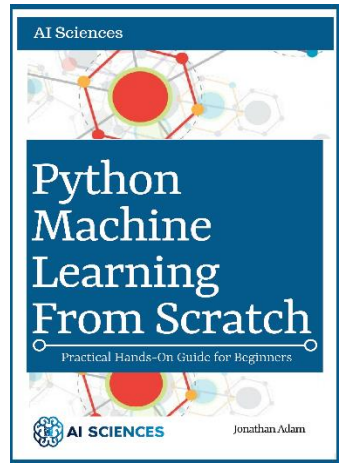
- Website: www.aisciences.net
- Email: contact@aisciences.net

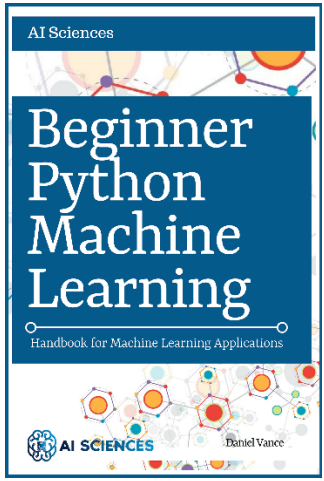
Follow us on social media and share our publications

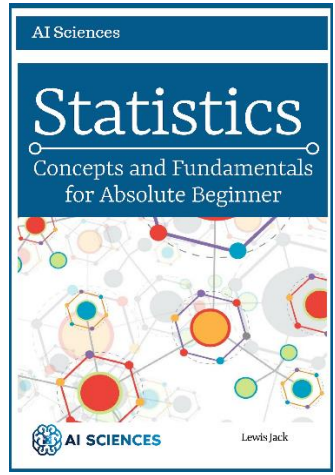
- Facebook: **@aisciencesllc**
- LinkedIn: **AI Sciences**

From AI Sciences









WWW.AISCIENCES.NET

EBooks, free offers of eBooks and online learning courses

Did you know that AI Sciences offers free eBooks 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 in your career and 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 course does not throw complex maths at you. Instead, it focuses on building up your intuition for infinitely better results down the line

AI SCIENCES
ACADEMY
AI SCIENCES LLC

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

Preface

Why are the AI Sciences Books different?

The AI Sciences Books explore every aspect of Artificial Intelligence and Data Science using computer Science programming languages such as Python and R. Our books may be the best one 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, and learning any other high-level courses will be easy for you.

Who This Book Course Is For

This book is designed for students and learners who desire to demystify 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 basics, and then to gradually develop an understanding of the array of machine learning algorithms. This book emphasizes on practical implementation along with a theoretical explanation of the concepts.

It is better to have Python, statistics and math understanding before starting this book. However, if you're new to the subject, it is not a limitation.

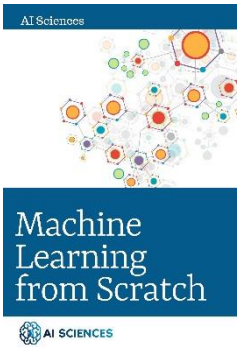
Why this book?

This book will guide you step by step from the very basics to what Machine Learning is. The best part about this book is its

structure, it's structured in such a way that make the concepts easily understandable. It will help you to understand a basic about Machine Learning and master it in ONE DAY! Thus ensures 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 will not only guide you through the problems and concepts of Machine Learning but also elaborates how one can successfully use it to implement the concepts.

Your Free Gifts

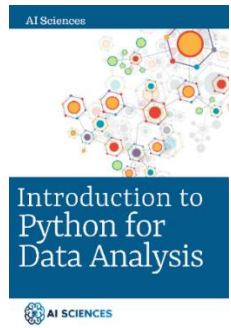
As a way of saying thank you for your purchase, AI Sciences Publishing Company offers you a free eBook in Machine Learning with Python written by the data scientist Alain Kaufmann.



It is a full book that contains useful machine learning techniques using python. It is a 100 page book with one bonus chapter focusing on Anaconda Setup & Python Crash Course. AI Sciences encourages you to print, save and share. You can download it by going to the link below or by clicking in the book cover above.

http://aisciences.net/free-books/

AI Sciences Publishing Company offers you also a free eBook in Introduction to Python for Data Analysis written by Robert Danboard.



It is a full introduction book in Python for Data Analysis. It is a 100 pages book. AI Sciences encourages you to print, save and share. You can download it by going to the link below or by clicking in the book cover above.

http://aisciences.net/free-book-3/

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

Chapter 1: Introduction to Machine Learning

What Is Machine Learning?

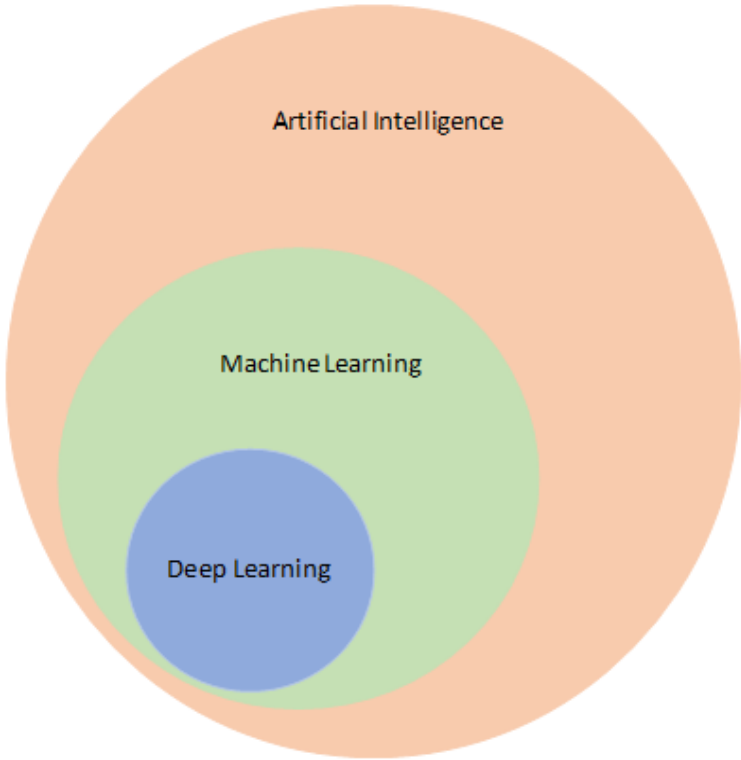
Many of the current computer systems use fixed logic to interpret input data and calculate the output. With continuously changing world it is difficult to maintain these systems and accuracy of output is also become questionable because the data for which the logic was built has drastically changed over a period. Use of internet devices is increasing day by day, and these devices are generating a humongous amount of structured and unstructured data every hour. Old computer systems with static rules are incapable of processing this data and extract knowledge out of it. Availability of computation power and Machine Learning algorithms are paving the path to building 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 which observed positions at the game and learned a model which gave better moves for the machine player. The program was able to play better over time by getting more experience from more games it 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

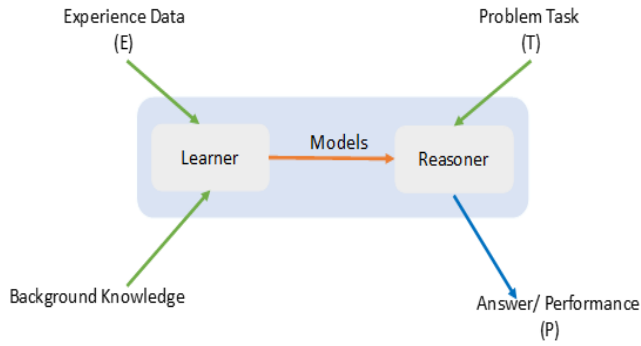
generalizing from historical data and can learn to improve themselves from the experience of past data.

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



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 includes a 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 right, the model will not get proper knowledge from the data and would implement wrong learning in calculating the output. Hence it is one of the crucial steps in building Machine Learning models. Finalizing a right dataset is an iterative process with many loops. Data preparation is 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 the format which Machine Learning model will accept. Removing noise, handling missing data, handling outliers, etc. would help to prepare the dataset for good model learning.

- **Preprocess Data—**Selected data may not be in a suitable format to work with. In that case, it needs to be converted into the format which 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. 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 training and test dataset. We use a training dataset to train the Machine Learning model, and we use a test dataset to validate the accuracy of output. Generally, the split is taken ~20% test data, but it is not mandatory to stick to only 20% as a test dataset. It may vary as per the data at hand. It is important to have correct training dataset because

the initial learning of Machine Learning model will be from this only.

- **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 strategy needs to be chalked out to migrate it into the path to production as well as how it will evolve with time. How frequently the Machine Learning model needs to be retrained so that it can calculate the output correctly.

Problems that Machine Learning Can Solve

In the recent past, machine learning algorithms are accepted in a range of industries and consumer areas. Adaptive learning and continuous improvement enabled Machine Learning to play a key role in the evolution of commercial,

social and educational domains. Machine learning has become part of the day to day life. Be it getting an automatic recommendation for which videos to watch, which product to order, what food to eat, to fingerprint and face recognition, tagging friends in a digital photo; machine learning has become a backbone of many websites and devices. A successful machine learning solution learns from generalizing input data and predicts accurate outputs for the input data which it has never seen.

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

Medicine

- Learn from historical medical records to learn which patient will respond to which treatment.
- Disease diagnosis—Data such as symptoms, lab measurement 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

- Digitizing handwritten scripts
- Number plate identification of moving the car
- Face recognition—Face detection and unlocking a mobile device

- What an object represents in an image and where it is presented
- Automatic 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 Robot—autonomous robots which 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.

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 the customer input (text or speech), understand the context and reply with 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

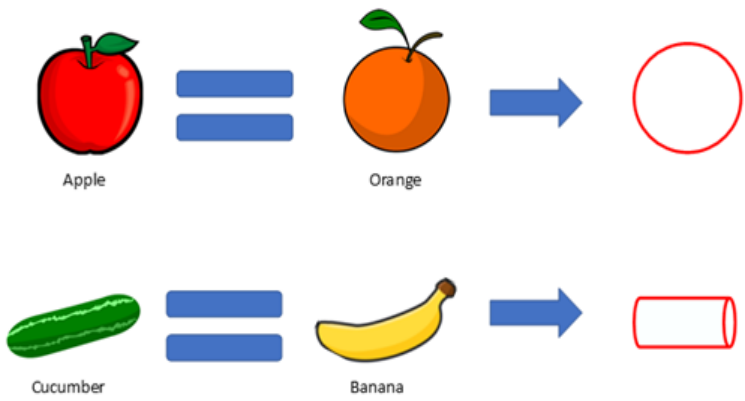
- Take a data-driven approach to identify how the climate is changing, accurate forecast of cyclones, earthquakes, hurricanes, etc. will help in saving lots of lives.

- Weather Forecast—There are many industries which cannot function if it is raining or it is too hot or too cold. The 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's 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 fruits are same, be it orange, apple, banana, cucumber or any other fruit for that matter. In his first phase of learning, he builds an intuition that orange and apple are of one shape, and banana and cucumber are of another shape.

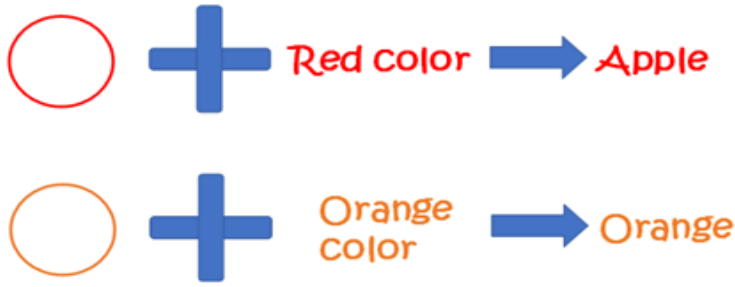
Phase 1 Learning



Once the baby is comfortable with shapes of the fruits, he is introduced to another property of the fruit, e.g., color. Now he knows that round shape & red color fruit designate an apple and round shape & orange color fruit belong to an

orange. Similarly, now he would be able to distinguish between a banana and a cucumber.

Phase 2 Learning



Now the baby can clearly distinguish between round shape and cylindrical shape fruits. But to reach to this stage baby was told numerous times that the round shape & red color fruit is an apple, and the round shape & orange color fruit is an orange. If we put a subset of the information given to the baby in a tabular format, it will be like this:

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 of the fruit: shape and color. And the baby was told about combinations these properties again and again until he learned how to identify the fruit type as per these properties.

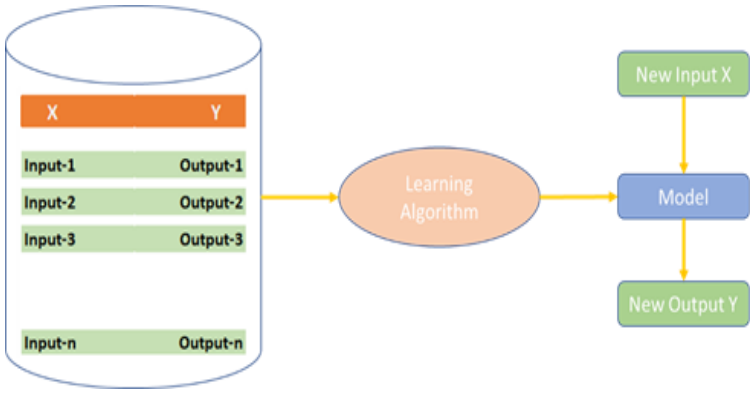
Machine learning models learn in the same way. In machine learning, the properties of the fruit, such as shape and color, are called the **features**, fruit type is called the **label**, and each instance of input-output pair is called an **observation**.

		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 features and label passed to a machine learning algorithm, learning is classified into four categories.

Supervised Learning

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



Supervised learning is also referred to as "Learn from the past to predict the future."

Supervised learning algorithms receive input features and corresponding correct output. With each iteration, the algorithm learns by minimizing the error between correct output and the predicted output. To minimize the error, the model is modified by the algorithm in 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 types of machine learning so far. The downside is that it often requires human effort to label the training data set. 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 to one or more of these classes. When the learning dataset label has exactly two categories, it is called binary classification, and if learning dataset label 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, and in a class examination grading student in A, B, C and D grades is multi-class classification.

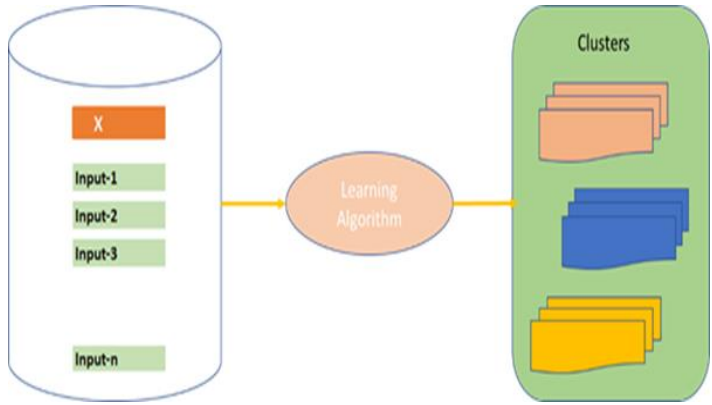
  In binary classification one class is termed as positive and another class as negative. Here positive class doesn't represent profit or benefit but rather represent the object in the study.
- **Regression—**The learning dataset label contains real numbers, and the Machine Learning algorithm produces 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.

  E.g., creating a model to predict house prices is an example of regression. Here learning dataset will have multiple observations along with a real number assigned as house price for each observation. The output of this regression model will also be a real number as 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 to so few training examples. As a result, it's hard to learn what a fraudulent activity looks like. The approach that anomaly detection takes is to take a history of non-fraudulent transactions to determine what normal activity looks like, and then to identify anything that is notably different.

Unsupervised Learning

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



Unsupervised learning algorithms get 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 learning dataset. There are many successful applications of unsupervised learning

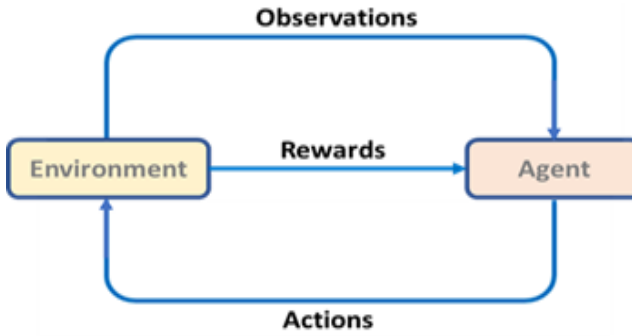
models. However, these are harder to understand and evaluate. Unsupervised learning is useful in market segmentation, product recommendation, etc.

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 models will recommend what other product you may like according to the product you searched for. The model can do it because the model created clusters as per 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

It is often used in robotics, gaming, and navigation. With reinforcement learning, the algorithm identifies a policy (strategy) to get a maximum reward in short and longer term. The policy decides which 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 do an action in the environment. This agent's action takes the environment to a new state and environment gives a reward to the agent. This reward can be a negative reward or a positive reward or a penalty or nothing. From multiple iterations, the agent learns a policy of what action to take in a given state of the environment so that not only short-term reward is optimized but the overall utility of the agent in a given time horizon is optimized.

Reinforcement learning became popular in March 2016, when AlphaGo program beat world champion Lee Sedol in a game of Go. The program analyzed millions of games 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 dataset. Semi-supervised learning is used for the same applications as supervised learning. It is useful when collecting 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.

Commonly used application of semi-supervised learning is tagging friends on Facebook or Google. Once you upload multiple photos of the same person, the algorithm clusters them according to the face 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 photo, all photos in cluster one will be labeled as "person A."

Instance-Based Learning

Till now, the learning methods we discussed are considered as model-based learnings. In model-based learning 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 learner never goes through 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. The delay in training dataset processing leads to more time consumption during the prediction phase.

The key advantage of instance-based learning is that it is more dynamic compared to 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 future based on historical events. In this chapter lets discuss about math required to understand the machine learning algorithms. This understanding will help in model evaluation and why one model may behave better than the other one in a certain scenario.

Machine learning is powered by Linear Algebra, Calculus, and Probability and Statistics. Let's discuss them one by one.

Notation

Machine Learning involves lot of mathematics 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. Most commonly used Machine Learning notations that we will use in this book are-

	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	
	ρ_X	correlation	correlation of variables X and Y	
	$\tilde{x}$	median	median value of variable x	

Data Structure

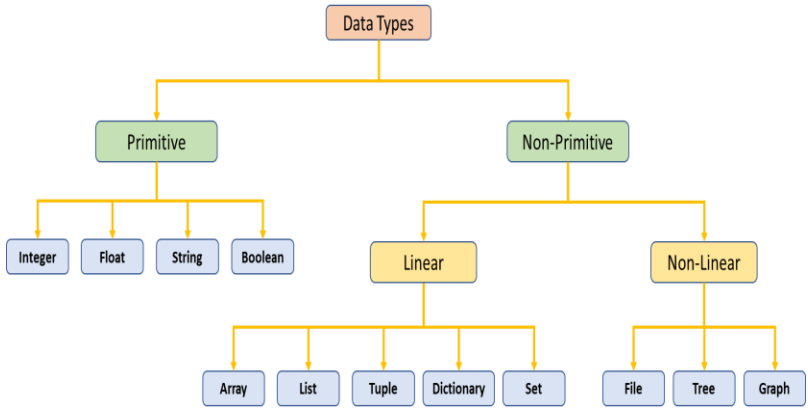
Till now we have discussed a lot about the data. But in what structure the data should be in?

Data Structure is an organized form of representing and storing data, which helps in 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 appropriate way.

Data structures are mainly grouped in two types-

1. **Primitive data types**- These are the core data types which 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, False
	String	>Sequence of characters >Immutable	'My name is Alex'
	Array	>A list of elements of same data types >Can have one or	$\begin{bmatrix} 2, 5, 7, 9 \\ 3, 6, 8, 4 \end{bmatrix}$

non-Primitive		more dimensions >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 can't be changed. E.g. values of variables t1 of data type Tuple can't be changed. It's a immutable data type.

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 lets discuss about Sets, Vectors and Matrices.

Set: - Set is defined as an unordered collection of unique elements. Sets can have heterogeneous elements, but one element cannot repeat again in the same set. Sets are declared as **auto = {1, 4, 6, 9, 'a', 'c'}.** Here auto is the name of the variable of type set, and it has 6 elements, four numbers 1, 4, 6, 9 and two characters 'a' and 'c'.

Operations on sets

Let's consider following two sets to understand operations on sets.

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

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

Union (∪)- Union of two given sets is the smallest set which contains all the elements of both the sets. The union of two given sets A and B is a set which consists of all the elements of A and all the elements of B such that no element is repeated. The symbol for denoting union of sets is **'∪'**.

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

Intersection (∩)- Intersection of two given sets is the set which contains all the elements which are common in both the sets. The symbol for denoting intersection of sets is **'∩'**.

s1 ∩ s2 = {1, 4}

Difference- Difference of s1 an s2 is a set of elements which are present in s1 but not in s2. It is denoted with symbol '−'

s1 − s2 = {6, 8, 12, 18, 23}

s2 − s1 = {3, 5, 9, 19, 43}

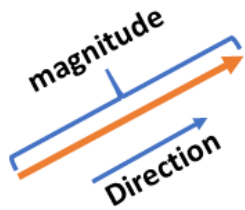
Symmetric Difference- Symmetric difference of s1 and s2 is a set of elements in both s1 and s2 except those that are common in both. It is represented with symbol 'Δ'

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

Complement- Complement of set s1 is a set of all elements except those which are present in set s1

Vectors

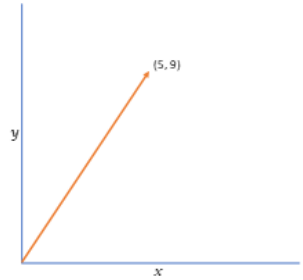
In physics scalars are defined as something which has value (magnitude) but no direction associated to it. And vectors are defined as something that has both magnitude and direction. Where magnitude is length of the vector and orientation is direction of the vector. e.g. speed is scalar since it does not have direction associated to it, but velocity is 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 number space. This point could be in a two dimensional, a three dimensional or a n dimensional space.

e.g. a = [5, 9]



Each element in a vector has an index associated to it. In above vector Element '5' can be accessed as a[0] and 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 index associated with it. In above example element '-8' can be accessed as $X[1][2]$

If a matrix has same number of rows and columns, then it is called a square matrix. E.g above matrix X is a square matrix

Operations on Vectors and Matrices- let's consider following vectors and matrices to understand operations on vectors and matrices

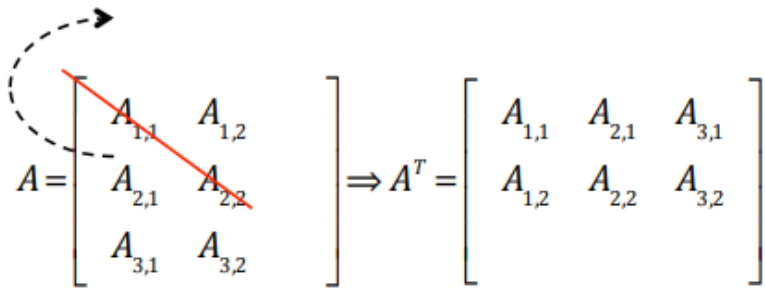
v1 = [6, 8, 13, 45, 32, 6]

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- Transpose of a matrix is defined as mirror image across a diagonal line. Transpose of a matrix A is denoted as AT

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



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

Addition- Vector addition is possible only if size of both the vectors is same.

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

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

Matrix addition is possible only if size of both the matrices is same.

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

Subtraction- Vector subtraction is possible only if both vectors are of same size

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

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

Matrix subtraction is possible only if size of both the matrices is same.

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

Comparison- 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 same size vectors

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- It can be done between two equal length vectors. 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 if no of columns in first matrix should be equal to no of rows in second matrix. If matrix **A** is of order (m, n) and matrix **B** is of order (r, s) the A X B will be possible if n=r. And the resultant 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

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 square matrix of order n where all the diagonal elements are '1' and all the other elements are '0'. It is denoted by ***I.***

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

Multiplication of a matrix with its inverse is an identity matrix. When a vector is multiplied with an identity matrix it does not changes its value.

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

Functions

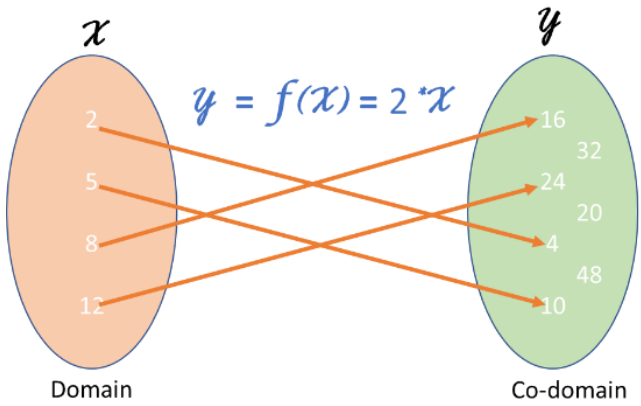
Function is a relation between an input and an output where each input is related to one and only one output. The set of allowable input values is called, domain and the set of all possible output is called, co-domain of the function. Set of output values corresponding to each domain value is called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

In the below example

domain = [2, 5, 8, 12]

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

range = [16, 24, 4, 10]



In Machine learning all the all algorithms use matrix multiplication for function calculation.

e.g. A function is defined as $y = a + a_1X_1 + a_2X_2 + a_3X_3 + \ldots\ldots\ldots\ldots\ldots\ldots + a_nX_n$

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

$$\begin{bmatrix} X_{11} & X_{21} & X_{31} & \ldots\ldots\ldots & X_{n1} \\ X_{12} & X_{22} & X_{32} & \ldots\ldots\ldots & X_{n2} \\ X_{13} & X_{23} & X_{33} & \ldots\ldots\ldots & X_{n3} \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ X_{1m} & X_{2m} & X_{3m} & \ldots\ldots\ldots & X_{nm} \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ . \\ . \\ . \\ . \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ . \\ . \\ . \\ . \\ y_m \end{bmatrix}$$

Derivative and Gradient

Rate of change- It is a measure of, how a variable is changing with respect to another variable. For an example, if you are travelling to your home by a car, then 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 states what is the average of rate of change for a function. E.g. distance between your home and house is 40 kilometers and it takes you 40 minutes to reach home. Then average rate of change in distance (speed) is

40 KMs / 40 mins = 1 KM per minute

Instantaneous rate of change- This measure states what is the rate of change at a position or moment. E.g. on your way to 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

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

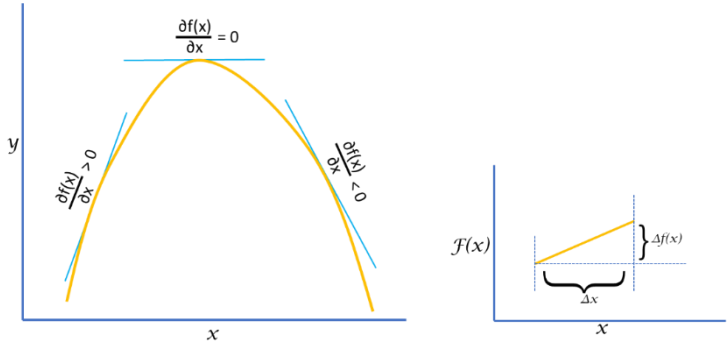
Δt is very small change in time and approximately equal to zero.

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

It is denoted as

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

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



Derivatives of commonly used function types are –

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- It gives the rate of change of a function in every direction (= no of variables). It has both magnitude and direction hence presented as a vector. It helps in calculating the slope at a specific point on a curve for functions with multiple independent variables. It points in the direction of 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

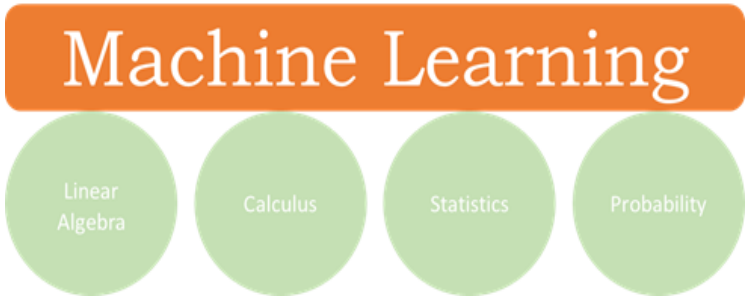
$$\nabla f(x,y) = \left(\frac{\partial f(x, y)}{\partial x} , \frac{\partial f(x, y)}{\partial y} \right)$$

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, Statistics and Probability. In this chapter, we will discuss Statistics and Probability and what role they play in 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] Let's understand 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 a business an online cloud selling company. You want to send a campaign to customers, but you might have of questions like, to whom you should send the campaign and to whom you should not? To how many

customers you should send the campaign? How effective would the campaign be? Will the person who receives the message buy more than the person who doesn't? Statistics is useful in getting the answers to these questions.

Statistics is classified into two types.

Descriptive Statistics

It is a method of organizing, summarizing and presenting data in an informative way. Descriptive statistics helps in understanding the data and getting insights from it. For example, you want to know how many customers are coming to a store, how many of them are female, how many of them are male, how many of male customers are smokers, how many of female customers are not smokers, how many of them are married. Descriptive statistics can answer all these questions.

Inferential Statistics

Many times the collection of an entire data set (population in statistics) is impossible. Hence a subset of the population is collected, also called sample, and a conclusion about the entire population is drawn. Conclusions about the population data set are inferred from conclusion of the sample data set. For example, let's imagine that elections are going to happen in a state, and we want to know which party is going to win. We cannot ask each person in the state of their opinion about which party will win this time. We will ask a few people, 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 whom we asked about their opinion.

Before jumping into the ocean of statistics let's define some generic terms used in statistics.

Introduction to Basic Terms

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

Population

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

The population is further categorized into two classes.

1. Infinite population – The data set where all data points cannot be counted, such as the stars in the universe.
2. Finite population – The data set where all data points can be counted, like, for example, all subscribers of a telecom company.

Sample

A sample is a subset of the population. If we select a few stars from the population (universe) for study, then a 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

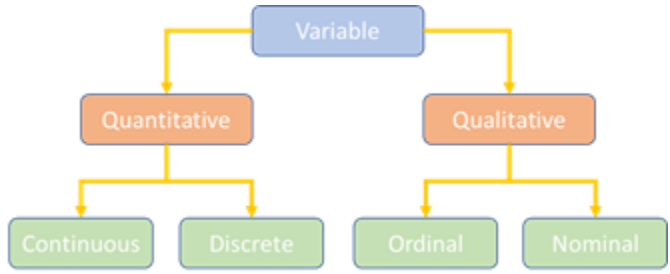
A variable is any characteristics, number or quantity that can be measured or counted. For example, if we want to run a campaign for customers, we want to know how many of the customers are adults, how many of them have kids, how many of them are married, what is their background, etc. All these are variables that help in understanding and analyzing 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, average age of students in a university, etc. Quantitative variables are divided into two categories.

- Discrete—These are whole integer numbers; e.*g., 5 (not 5.5) customers can come to an ATM.*
- Continuous—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 color. Male or Female etc.* Qualitative variables are further categorized into:

- Ordinal—Discrete values which can be compared. E.g., the rating that is given to a cab driver by two passengers.
- Nominal—Discrete values which cannot be compared. E.g. Male or Female, Yellow or Blue etc.



Data

Refers to values collected for the variable from each of the elements belonging to the sample or population. *E.g., a person carries a phone of brand X, 5 customers purchase $50 gift cards, etc.*

Experiment

A planned activity with results that yield a data set, e.g., where an advertisement should be displayed to generate more sales, whether on a website, or on social media app, or in a newspaper, etc.

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 USA, 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

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

Statistics

It refers to a numerical value that summarizes the sample data. E.g., to calculate the average salary of all males in the USA, we collect the average salary in different states. The average salary of each sample is statistics.

Measures of central tendency

(Mean, Median or Mode)—These are the parameters which attempt to describe a data set by identifying the central position within the data set. As such measures of central tendency are sometimes called measures of central location. Measures of central tendency help in comparing two data sets.

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

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

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

(18+ 21+ 19+ 20+ 18+ 21+ 22+ 19+ 18+ 20)/ 10 = **19.6**

- Median—A median is the middle value of the data set when the data points of the data set 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 two middle values.

In the above example, the median is:

Sorted values à 18, 18, 18, 19, **19, 20**, 20, 21, 21, 22

Median = (19 + 20) / 2 = 19.5

(Total No. of records is even.)

- Mode—A mode is the most occurring value in the dataset. *In the classroom example mode of the class is 18*.

Interesting facts about Mean, Median, and Mode

- If the data distribution is more towards large values, then these values intend to inflate the mean. In this

kind of scenarios, the median is not much influenced by large values, and 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. E.g., the mean sale of two stores is the same, but it does not tell 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 in getting these insights from the data set. Measures of dispersion characterize how the data is distributed. Commonly used dispersion measures are:

- **Range**—It is the difference between extreme values of a dataset. (Max value - Min value).

- **Variance—**Variance is a statistic that measures the closeness between data points in a data set. It is the arithmetic mean of squared deviations from the sample mean. It can be calculated using the below 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** – Standard deviation is a statistic that measures the dispersion of the data around the mean. It is the square root of the variance.

ChebbySheff's Theorem—It 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:

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

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

Probability – The probability of an event is a measure of the likelihood that the 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 happen.

Thumb rules 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, either a head or a tail. Since heads and tails are mutually exclusive events, they cannot occur simultaneously. Probability of coming head is (outcome head)/ (total no of possible outcomes) = 1/(1+1) = 0.5 and probability of coming tail is 1/ (1 + 1) = 0.5.

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

P (A) = (no of elements in A) / (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**

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

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

$$P(A \cap B) = P(A)\,P(B)$$

For example, tossing a coin and raining in California are two independent events. The probability of coming Head (A) and raining in California (B) will be **P(A) * P(B)**

- **Conditional Probability**—When a certain event has occurred then what is the probability of another event to occur. E.g., given the probability of a train running late, what will be the probability of you reaching late to office.

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

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

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

E.g., a card is pulled from a deck of cards, what is the probability of this card to be a heart given that the card which is pulled is of red color?

P(H|R) = P(H ∩ R)/ P(R)

Probability of getting Red color card P(R) = 26/52 = ½ = 0.5
Probability of getting a heart and Red color= 13/52 = ¼ = 0.25
Probability of getting a heart given it is a 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.

E.g., while walking in a garden, we find that grass is wet, and we want to find the probability that it was raining, and the sprinkler was not on.

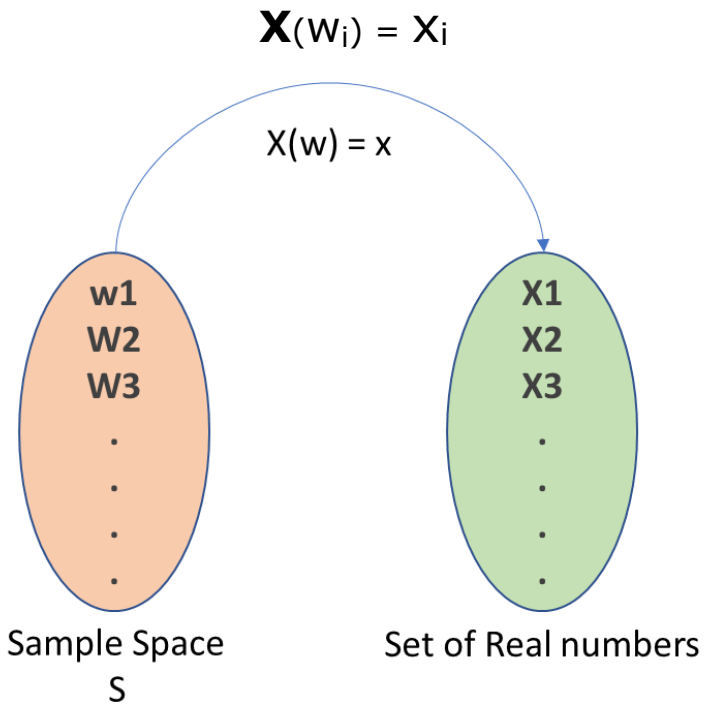
Reverend Thomas Bayes derived the formula below to calculate conditional probabilities.

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

Here:

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

Random Variable—Random variable is a function or rule which 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:

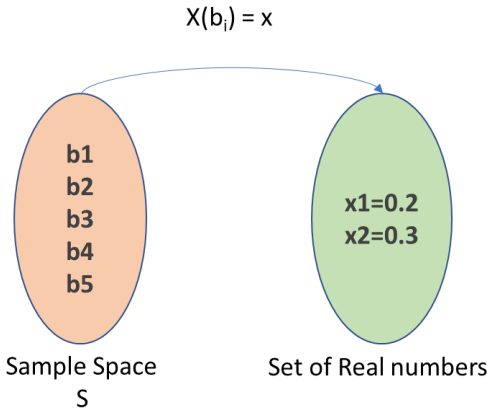


For example, let's suppose there are five balls, labeled as b1, b2, b3, b4, and b5, in a bag. The random variable X is the weight, in kg, of a ball selected at random. Ball b1 and b2 weight 0.2 kg and ball b3, b4 and b5 weight 0.3kg. This information can be represented as:

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 what all values it can take and what is the probability corresponding to each variable.

Average of the random variable is called expected value.

Random variables are classified into two types:

1. Discrete Random variable—A variable which holds only whole numbers. E.g., rolling of dice will produce one value in the range of 1 to 6. Probability distribution function in case of a discrete random variable is called a probability mass function.

For example, two dices are rolled simultaneously, and the sum of both the dices is a random variable. Possible outcomes are denoted as Value(X_i) and no of ways value can come is denoted with Frequency (n_i) and the probability of occurring Value(X_i) is denoted as Probability (P_i). i.e., total no of possible outcomes are (6*6 = 36), and value 2 can come in only 1 way (when both dice face 1) then the probability of

outcome 2 is 1/36. The probability distribution function for this scenario can be written as:

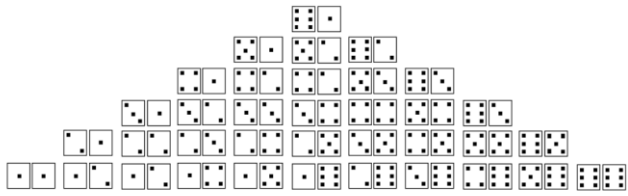
P(x= 2) = 1/36 P(x=5) = 4/36

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)—It is the cumulative probability of occurring one or more than one events. E.g., in the rolling of two dices example probability of $x < 5$ 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 – Variable which holds any real number. E.g., time taken in travelling from point A to point B. This travelling time could be 20 minutes, 2.5 hours or any other real number for that matter.

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

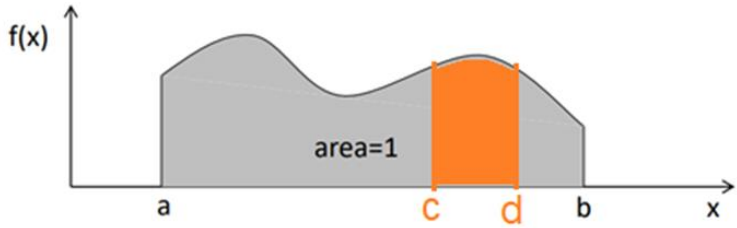
Consider an experiment where you are observing at the time interval between two trains at a metro station. This time interval can be any real number, i.e. 5 minutes, 5.01 minutes or 5.0123 minutes, etc. It is difficult to tell exactly how much interval will be there between two trains, hence for continuous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

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



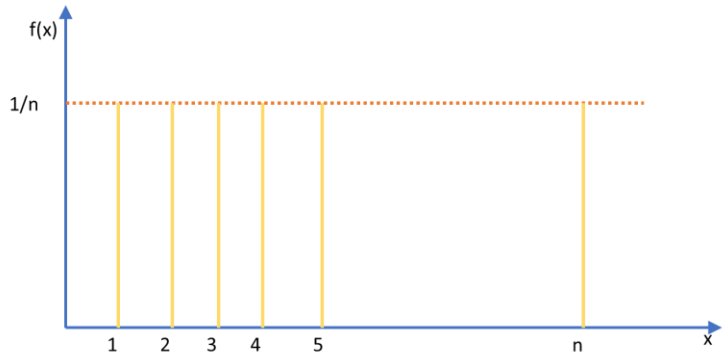
- Probability between c and d is the area under the curve between c and d (shaded region)
- P(c) =0 and P(d) = 0

Discrete Probability Distributions

Uniform

A distribution is a 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 ½.*

If no of possible outcomes (k) is known, all other parameters, e.g. mean and standard deviation can be calculated. No of 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: $P(X = x) = \frac{1}{k}$ $\quad (x = 1, 2, 3, \ldots, k)$

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

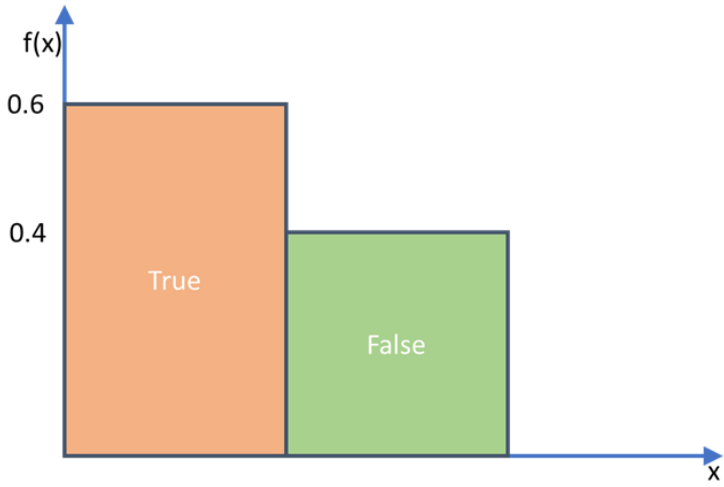
$$E[X^2] = \frac{(1^2 + 2^2 + \cdots + k^2)}{k} = \frac{\frac{1}{6}k(k+1)(2k+1)}{k} = \frac{(k+1)(2k+1)}{6}$$

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

Bernoulli

When the outcome of an experiment are only two values, True and False, then the distribution is called Bernoulli distribution. Here True and False means one of two possible outcomes of the experiment. *E.g., tossing a coin has only two outcomes, either head or tail.*

If the probability of occurring of True (p) is known, then all other parameters of the distribution, e.g. 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: $P(X = x) = p^x * (1-p)^{1-x}$, $x = 0, 1; 0 < p < 1$

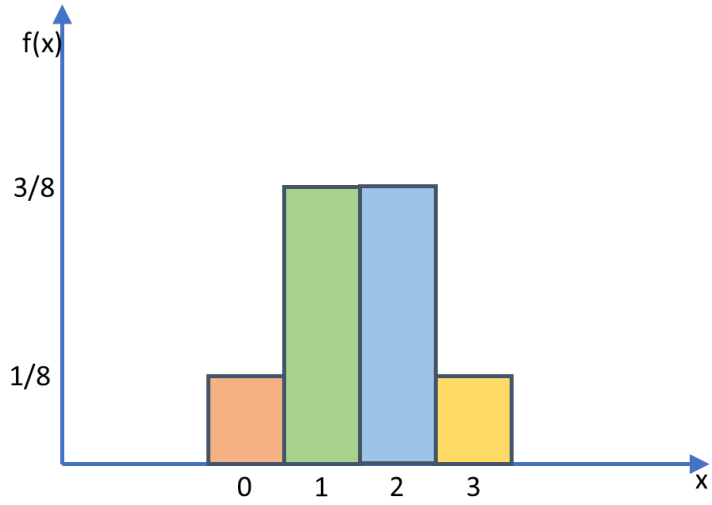
$\mu = p$

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

Binomial

When the Bernoulli distribution is repeated again and again it is called binomial distribution. E.g., 10 people visited my website. The probability that the first person would be subscribing for a newsletter is Bernoulli distribution. Similarly, whether the next person would subscribe or not is

again Bernoulli distribution. But if five people out of ten subscribes for the newsletter, it is binomial distribution.



If no of trials (n) the experiment was executed and probability (p) of success is known all other parameters of the distribution, e.g., mean and variance can be calculated. Hence **n** and **p** are called the parameter of Binomial.

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

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

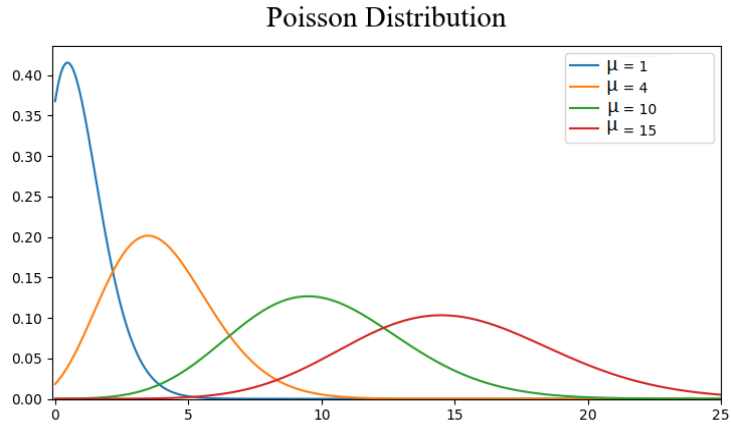
$$\mu = np$$

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

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

Poisson

It 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 assumptions are, the probability of success in an interval is same for all equal size intervals, and probability of success is proportional to the size of the interval. This distribution is helpful in traffic planning and ATM refilling planning etc. *E.g., if no of trucks passing USA and Mexico border is 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

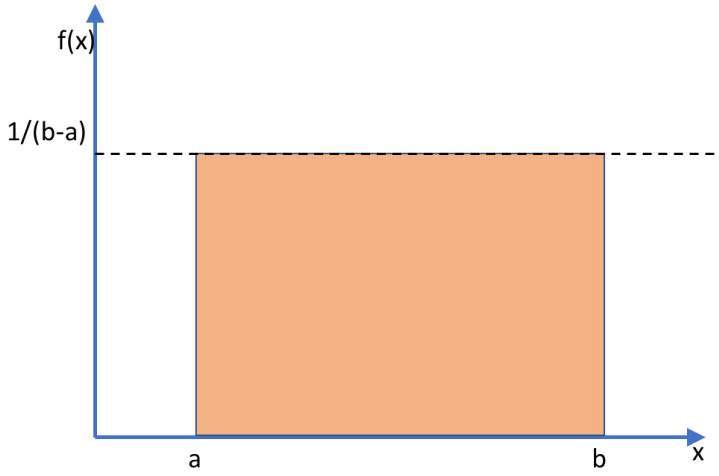
e – natural logarithm base

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

Continuous Probability Distributions

Uniform Distribution

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



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

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

It is a family of distributions, which are governed by two parameters **α** and **λ**. Change in any of these values will result in a different shape of the distribution. Later we will deep dive into a few special graphs from gamma family, e.g., exponential and chi-square distributions.

In below example for blue graph α =2 and λ = 3, and for red graph α =1 and λ = 4. Gamma distribution is highly used in insurance domain to predict claim amount.

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

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

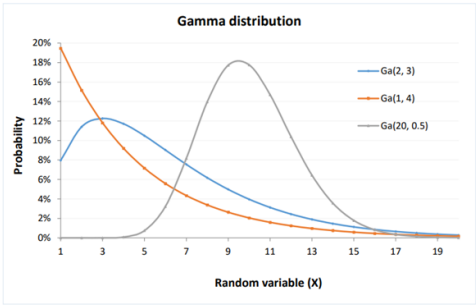
Variance, $\sigma^2 = \alpha/\lambda^2$

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 $\lambda = 1/2$

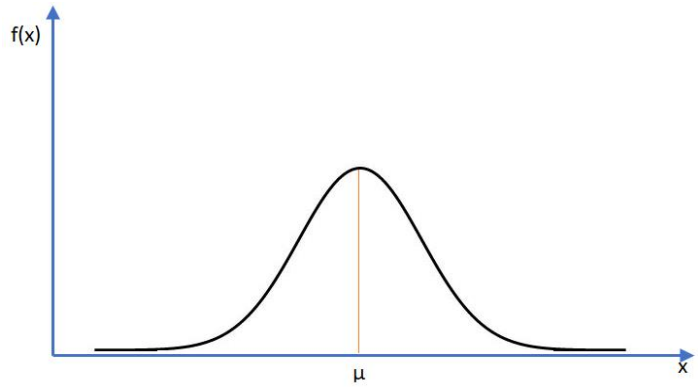
Plotting PDFs for different Gamma distributions using MS Excel.

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

The shape of this distribution is of a bell shape, and the distribution is symmetrical around mean (**μ**). The spread of the distribution is decided by the standard deviation (**σ**). Higher the standard deviation higher the spread. If the standard deviation is less than the distribution will be more shrunk. The peak of the normally distributed curve is kurtosis.

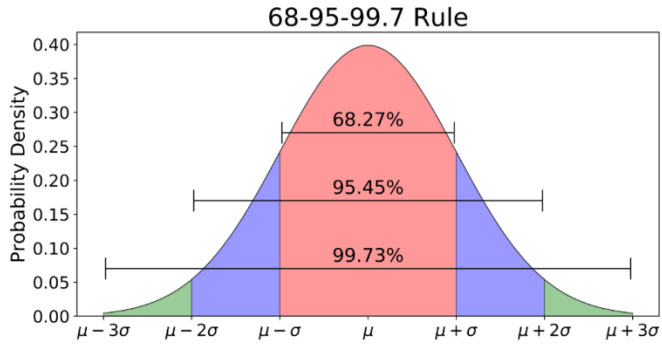


The probability density function of a normal random variable is:

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

The distribution of the data points in a normal distribution is as shown below. It means 68.27% of the data points will be in between (μ - 1σ) and (μ + 1σ). It means ~ 68% of the population lies between one standard deviation distance from the mean. Similarly, 95.45% of the population lies between

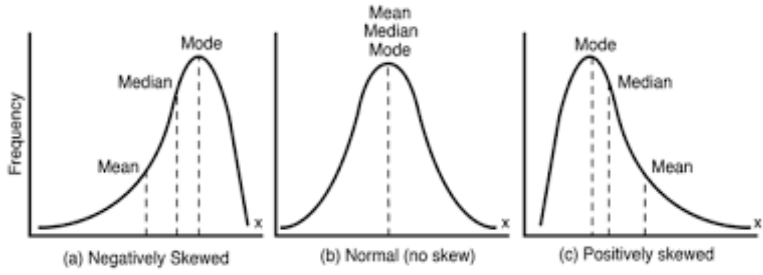
two standard deviation distances from the mean. It means that higher the distance from the mean, more population will be covered.



Skewness in the distribution

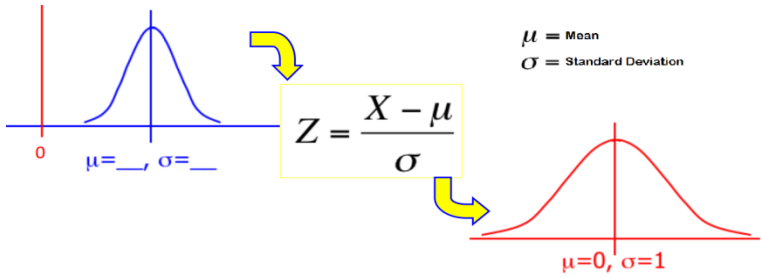
When a distribution is not symmetric, it is called the skewed distribution. When most of the data points are towards the right side of the distribution, it is called negatively skewed. For negatively skewed distribution mode is greater than the mean.

When most of the data points are towards the left side of the distribution, it is called positively skewed. For positively skewed distribution, mean is greater than the mode.



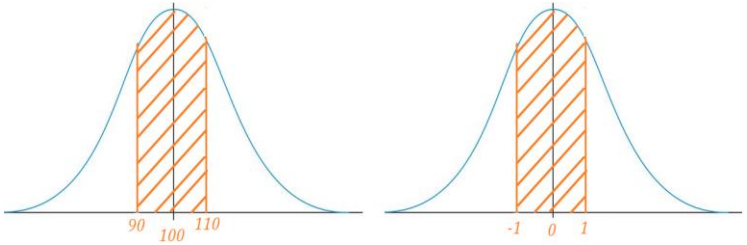
Standard Normal Distribution

It 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, mean = 25 and standard deviation = 10. Below formula is used to standardize all values of x.* For standard normal distribution mean (μ) = 0 and standard deviation (σ) = 1.



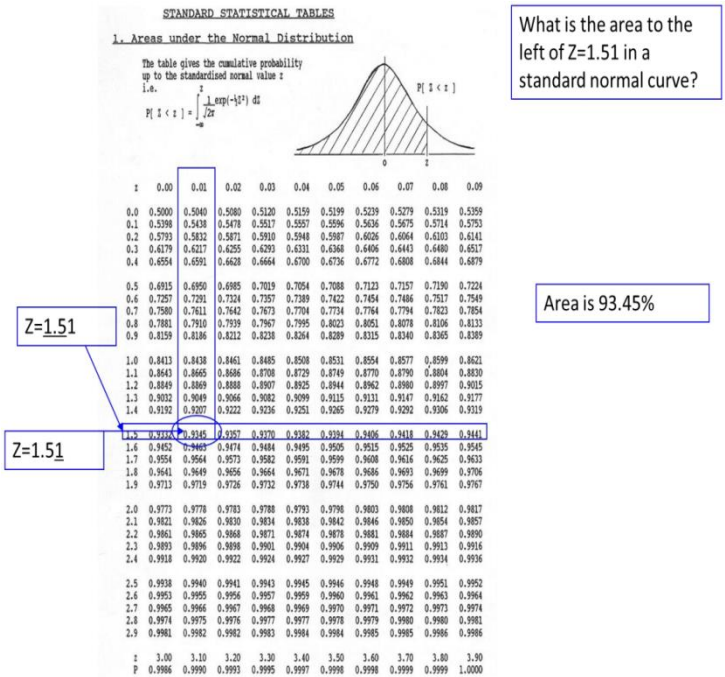
Standard normal distribution helps in calculating the probability (the area under the curve). Probability between two points in a standard normal distribution is the same as that of probability between corresponding points in a normal distribution. Probabilities for standard normal distribution are already available in the tabular format, which is known as the z table.

Let's look at an example. For normal distribution probability between 90 and 110 with standard deviation 10 and mean 100 will be same as that of probability between (90-100)/10 = -1 and (110-100)/10 = 1 with standard deviation 1 and mean 0. In the figure below, the shaded region (area under the curve) is the same.



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

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.

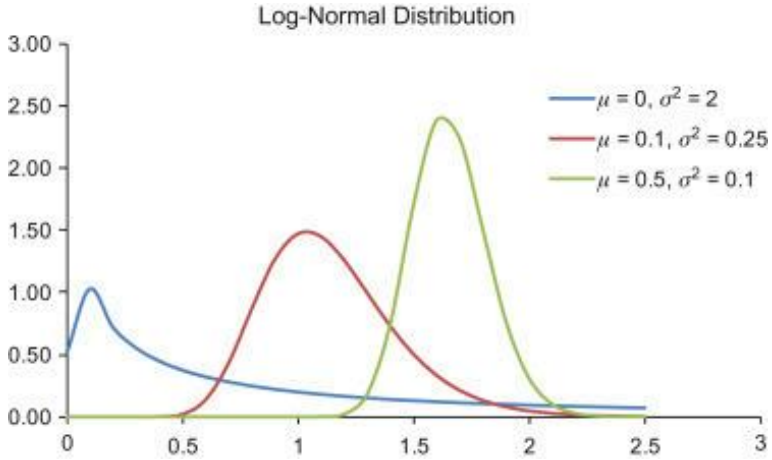
$$\mu_{\bar{x}} = \mu$$

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

$\mu_{\bar{x}}$ 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 called as lognormally distributed.



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

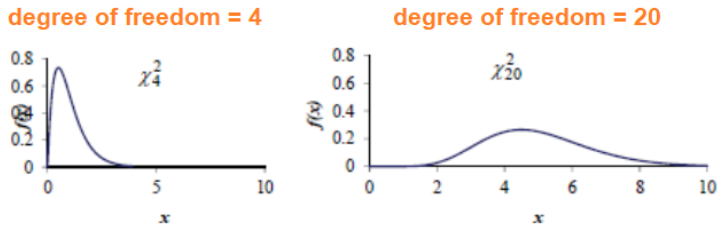
$$f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-\frac{(\ln x - \mu)^2}{\sqrt{2\sigma^2}}}$$

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

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

Chi-square Distribution

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



As we discussed earlier, it is very costly to go to a whole population and collect values. To counter that we collect values for sample data and infer conclusions for the population from the sample.

It is highly important to collect correct 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 difficult to go to everyone in the state and ask them for their age. We will rather select few people as a sample and calculate the average age of this sample. Similarly, we will select

multiple samples to make the inferences towards the whole population more generic. To make better inferences for the population, the sample chosen should represent the whole population.

In the above example, if the samples collected are of the elderly or young age people only, then the inferences made for the population from these samples are not correct.

Let's take an example to understand what a population would be and what would be a sample in generic use cases.

"Metrological department is studying 10 cities to establish whether air pollution levels are acceptable in UK cities".

In this example, the selected 10 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

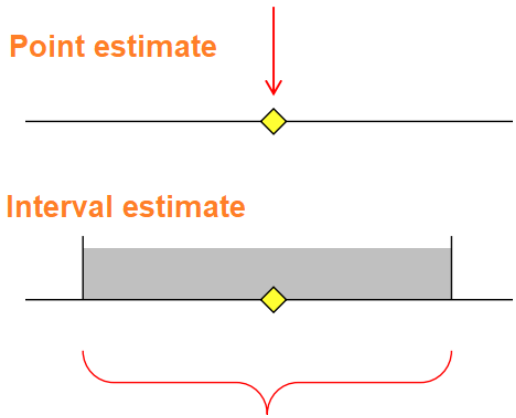
The objective of estimation is to get population parameters based on sample statistics. E.g., from the sample mean we try to estimate the population mean.

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., mean age of a college's students is estimated to be 21 years based on the sample mean of 50 students.* Estimating an exact value is difficult, and the chances of error are bigger.

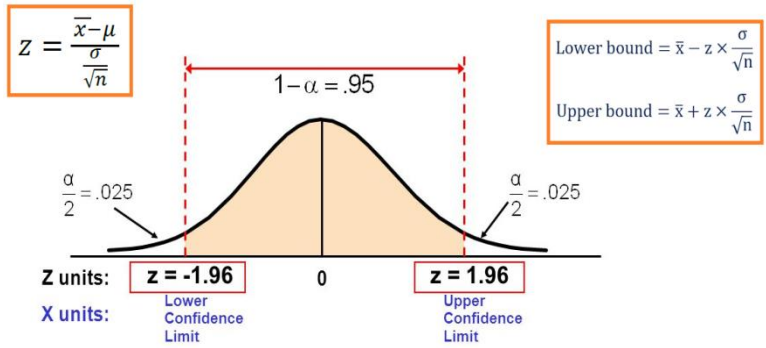
2. *Interval estimation*- When the estimation is a range or an interval of values then it is called interval estimation. *E.g., the mean age of a college's students is estimated to be in between 19 and 22 based on the sample mean of 50 students.*

With interval estimation, a probability can be assigned which indicates 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)

It is the confidence level of correctness of the estimates done for the population. Confidence level requirement will differ industry to industry. E.g., for a drug manufacturer confidence level might be 99.99% and for a milk packaging dealer confidence level can be 95%.



Commonly used confidence levels are:

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's try to understand the hypothesis with 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. Means reject the null hypothesis in favor of the alternate hypothesis.

2. Conclude that there is not enough evidence to support the alternate hypothesis. Means is not rejecting the null hypothesis in favor of the alternate hypothesis.

In hypothesis testing there are two possible errors:

1. Type-I error- It occurs when we reject a true null hypothesis. i.e., *when a jury convicts an innocent person.*

2. Type-II error- It occurs when we don't reject a false null hypothesis. i.e., when a jury acquits a guilty defendant.

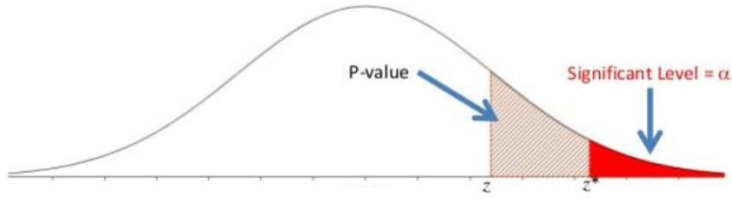
Probabilities of Type-I and Type-II errors are inversely related. Decreasing one increases others.

	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.

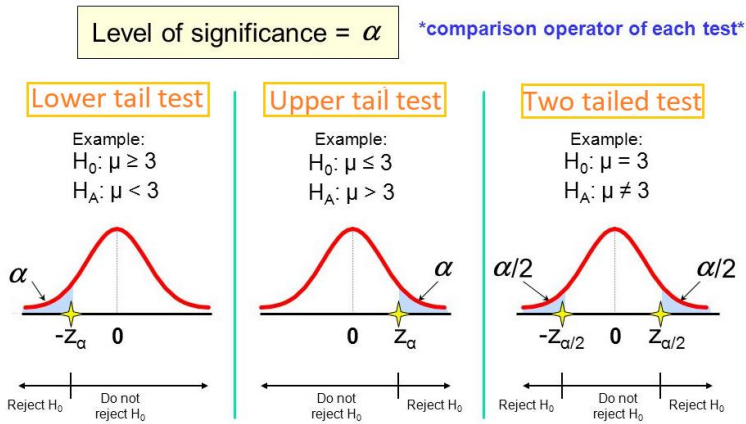
P-value test

It 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). Higher p-value means stronger support to the null hypothesis. In the below figure, the region shaded in brown is the p-value, and the region shaded in red color 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 means if the calculated Z test statistics is laying under 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 (the estimate should be correct with what 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 test or rejection region to make a decision.

Up until now, we have been discussing the calculating mean of the population when the standard deviation of the population is known. Now let's think for a moment, how the standard deviation of the population is calculated without knowing the mean of the population. 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 kinds of scenarios:

1. One sample t-test—It 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 the t statistics uses the estimated standard error.

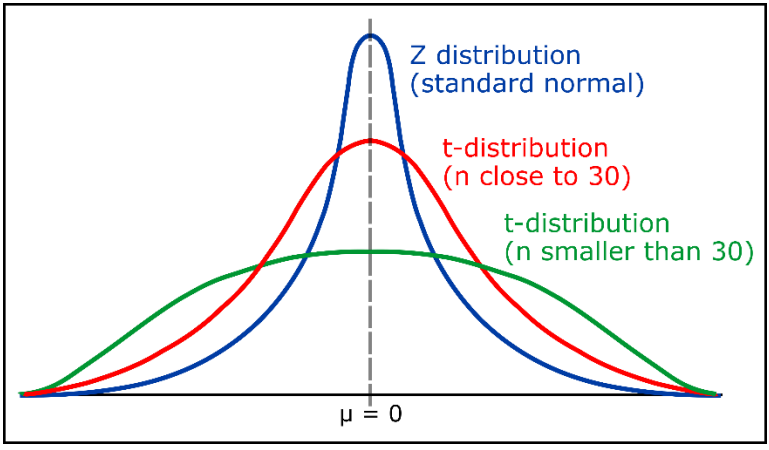
$Z = \frac{\overline{X} - \mu_{hyp}}{\sigma_{\overline{X}}}$	$t = \frac{\overline{X} - \mu_{hyp}}{s_{\overline{X}}}$
$\sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}}$	$s_{\overline{X}} = \frac{s}{\sqrt{n}}$
$\sigma = \sqrt{\frac{\Sigma(X-\mu)^2}{N}}$	$s = \sqrt{\frac{\sum(X-\overline{X})^2}{n-1}}$

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 remaining elements and the mean. So, if the mean is constant, there are only n-1 ways to play around with data. In simple terms, the degree of freedom is the number of scores in the sample that are free to vary.

Let's try to understand this with an example. There is a bag with 5 balls marked as 1,2,3,4,5, and the mean of the 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 the ball which is still there inside the bag by using the mean and the balls that are out of the bag.

Let's take another example. As the degree of freedom increases, the ***t*** distribution approaches towards a normal distribution. The degree of freedom increases as the sample size increased 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 when the degree of freedom's changes.



Steps for hypothesis testing

1. Formulate the hypothesis. E.g. 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

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

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 and based on these samples; we are trying to infer weather the two populations to which these samples belong to are same or not.

Steps for hypothesis testing:

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

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

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 same in both the samples, but the environment is different.

For example, a researcher would like to determine if the background noise causes people to take longer to complete math problems. The researcher gives 20 subjects two math tests, one with complete silence and one with background noise, and records the time each subject takes to complete each test. In this scenario, your test score is compared when you write the test with or without background noise.

Steps for hypothesis testing:

1. Formulate hypothesis. For example, 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, the population mean and how we can approximate population mean based on a sample mean given sample size is large or small, the standard deviation of the population is known or not. The next thing we'll talk about is the 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—It is denoted with 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.

The chi-square distribution with 1 degree of freedom:[vi]

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

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

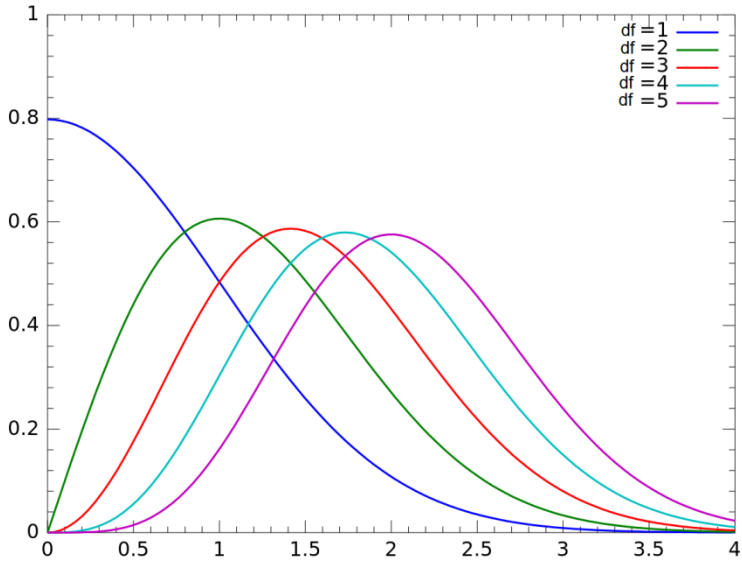
$$z^2 = \chi^2_{(1)} \longrightarrow \text{Make it Greek}$$

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 gets large, the curve is less skewed and more normal.



The characteristics of chi-square distribution:

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

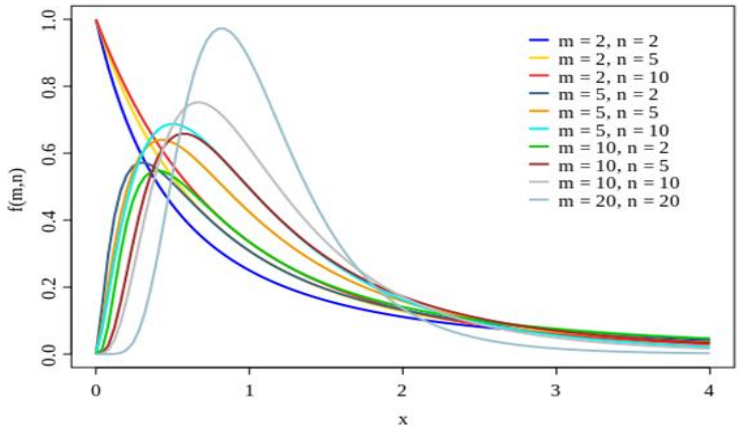
$$s^2 = \frac{\sum (y-\bar{y})^2}{N-1}$$

We can use the info 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. 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.

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

F-distribution—It is a family of curves, and each curve is defined by two degrees of freedom. F distributions are positively-skewed distribution. 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. 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, and 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.

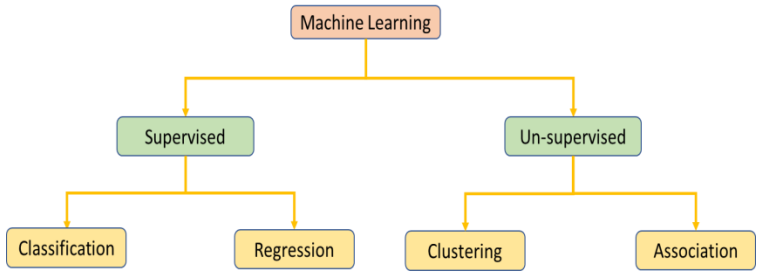
Machine learning and deep learning algorithms use probability distribution functions and statistics for predictions and classifications.

Probability distribution function and statistics are the back bone of machine learning. Many of the above statistical approaches are used in exploratory data analysis and feature engineering. E.g., for a house prediction problem, we want to know how a house with 3 bedrooms is different from a house with 4 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 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 the data which have labeled target. *E.g., if a model is being created for predicting house prices then the historical data which is used to train the model should have a target column stating the price of a house.*

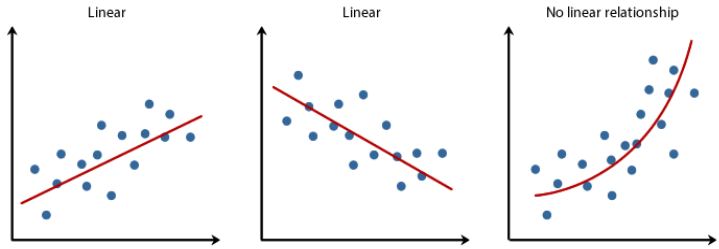


Supervised machine learning problems are broadly classified into two categories: regression and classification.

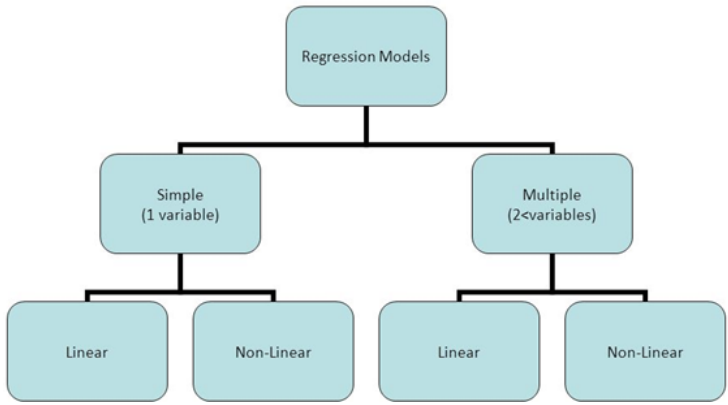
Regression

We use this form of predictive modelling 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 relationship 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 the simple regression. If the target is dependent on multiple predictors, then it is called the multiple regression.

In regression analysis, labeled historical data is available. Machine learning algorithms collect insights from this data

and use these insights to predict dependent variable (target) for new instances of an independent variable (e.g., predicting a house price in the USA, predicting marketing expenditure to boost sales of a certain product, etc.)

Classification

These algorithms are used to predict the target which has discrete values (known as target classes, e.g., orange, pineapple, and sweet lime are different classes of fruits). In this kind of use cases, 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 class, predicting whether a patient is suffering from cancer or not).

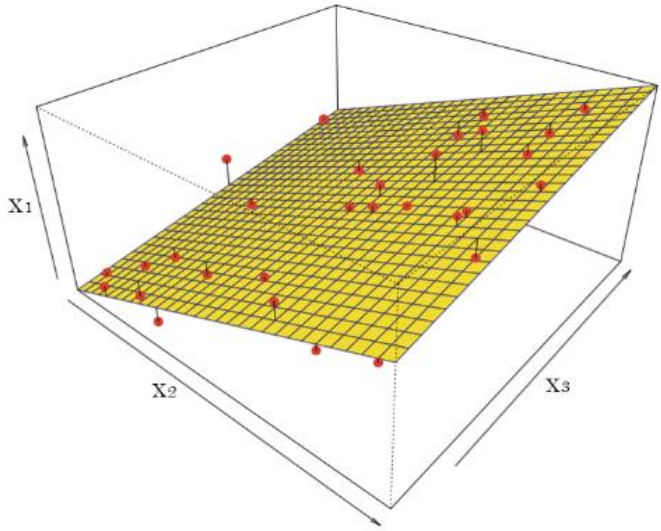
Commonly used regression machine learning algorithms are Linear Regression, Decision Trees, Support Vector Machines (SVMs), Naïve Bayes, Nearest Neighbor, Neural Network, etc. Let's discuss these algorithms in detail to build an understanding of how to use these algorithms and their pros and cons.

Linear Regression

It 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 which is continuous in nature and can be represented as a linear function of predictors/ features. *E.g., Predicting rental amount for a bike renting company, where bike rent can be represented as a linear function of weather, wind speed, etc.*

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. And the model is called a linear regression model. It means that a 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 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 hyper plane.

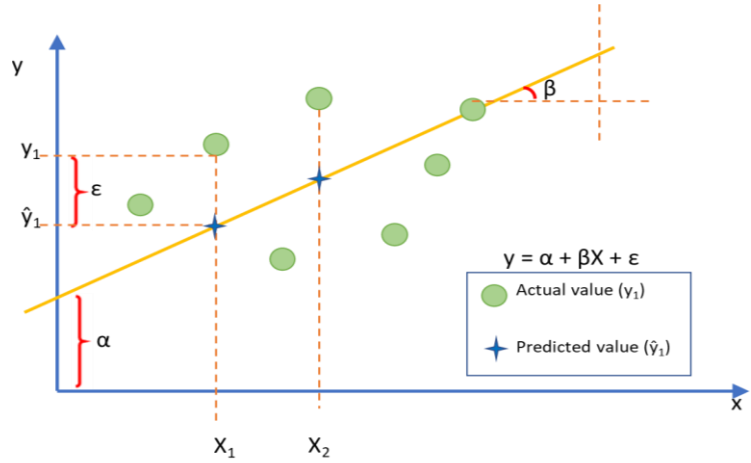


Simple Linear Regression

When the dependent variable (target) is dependent on only one independent variable (predictor) then the linear regression model is called as Simple Linear Regression model. And it 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.



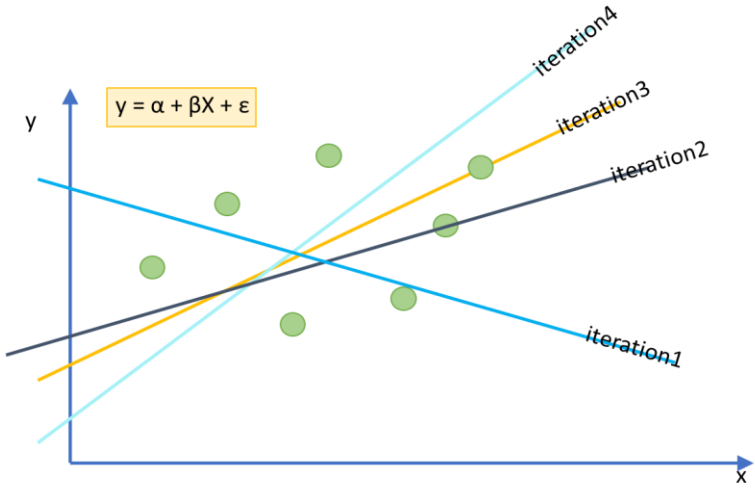
As shown in the above image, green color circles are original data points, and the orange color line is the fitted line (all predicted values of y lies on this line). Linear regression algorithm modifies α, β, and ε to find a best fit regression line so that the error between the predicted value and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

α and β are called model coefficients, and 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 learnt 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. In 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.

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

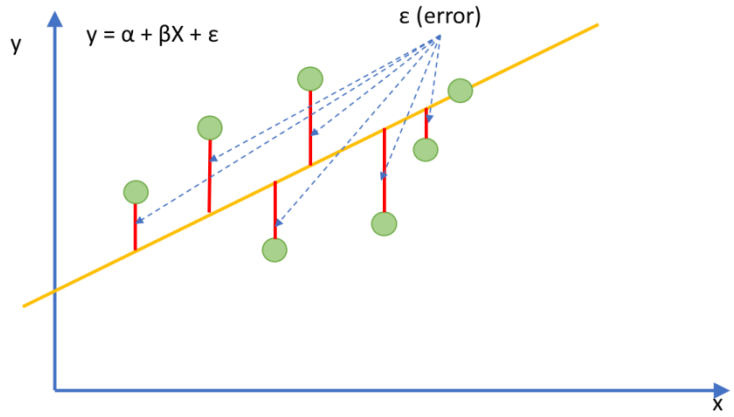
* n is the number of data points
* Y_i represents observed values
* $\hat{Y}_i$ represents predicted values



The error in prediction for each observation is represented as shown in the below image. In the below image red color lines are errors (ε), it is the difference between 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 cancelling each other. To avoid this in general practice squared error is used and it is calculated by squaring 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 mathematically it is 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 no 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 variable, and ε is the error between 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.

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

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

The above linear regression model is also known as ordinary least squares models (OLS models) 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 possibilities of improvement in accuracy.

MSE is the most commonly used error calculation method. However, other methods are also available for error calculation. Few 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 data set instead of being a generic solution). Hence the prediction accuracy of test data set is significantly low compared to the prediction accuracy of training data set. 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 Lasso Regression model, and linear regression model with L2 penalty is known as the 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 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, 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 of between accuracy and generalization. This trade of can be controlled via regularization parameter λ.

Regularization parameter (λ) and penalty term (L1 or L2) are the only hyper parameters for linear regression models.

Linear regression models are easy to understand and perform well for the sparse dataset. These models are fast to learn and predict. If the data set 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 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 data set 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 overcome this kind of scenarios highly correlated features should be removed from the dataset.

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

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

Benefits of linear regression:

1. Easy to interpret

2. Unambiguous and fast estimation

Downsides of linear regression:

1. Only mean values are estimated

2. Error (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 Mean Squared Error (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 which are categorical in nature. For example, we ca use logistic regression to determine whether a patient will respond to a certain medical procedure or not, a plant will survive or not, 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 dependent variable is categorical – it will either rain or not).

For a dependent categorical variable also, 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 is 0 or 1.*

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

Hence it is better to model for predicting probabilities for the dependent variable. But there is a constraint with the probability that probability cannot be less than zero and greater than *1. E.g., probability of raining 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. Odds $\frac{P}{(1-P)}$ in favor of an event is the ratio of probability it will occur to the probability it will not occur. Odds of the success represent the same information as that of the probability of success, but at a different scale. Probability is in between 0 and 1 with 0.5 in the middle. Odds are in between $-\infty$ and ∞ with 1 in the middle.

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

A linear relationship between probability of dependent variable and the independent variables can be established using various functions. However 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.

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

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

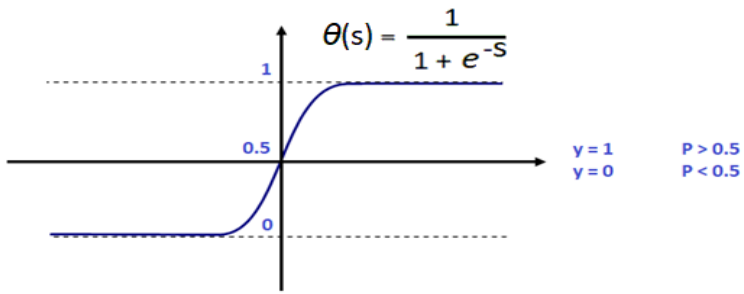
$$\ln\left(\frac{P}{(1-P)}\right) = S = \alpha + \sum_{i=1}^{n} \beta_i X_i$$

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

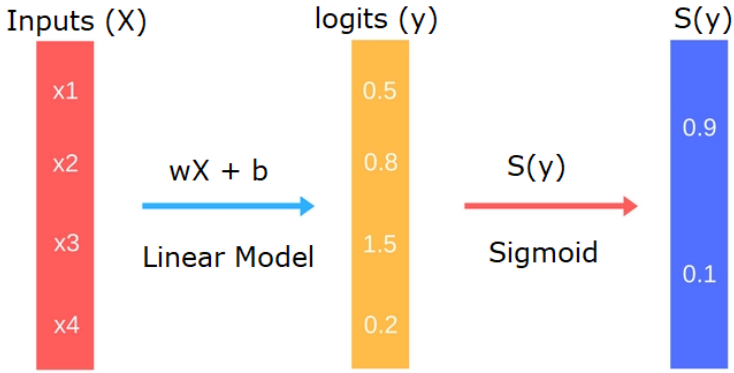
$$P = \frac{e^S}{1+e^S}$$

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

The probability of the occurring of an event is P and not occurring the event 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 if the predicted probability is greater than 0.5 then the category is predicted as 1 else 0.



Steps in logistics regression model can be depicted as below:



With this we want to learn a hypothesis $h(X)$ that best fits the above 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 data set D with N samples given a hypothesis (denoted arbitrarily 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-= 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 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 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 required twice 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 model coefficients vector W_{i+1} in the current iteration is the sum of model coefficients of previous iteration (W_i) and negative of the partial derivative of model coefficients of previous iteration ($\nabla E_{in}(W_i)$) multiplied by learning rate (η).

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

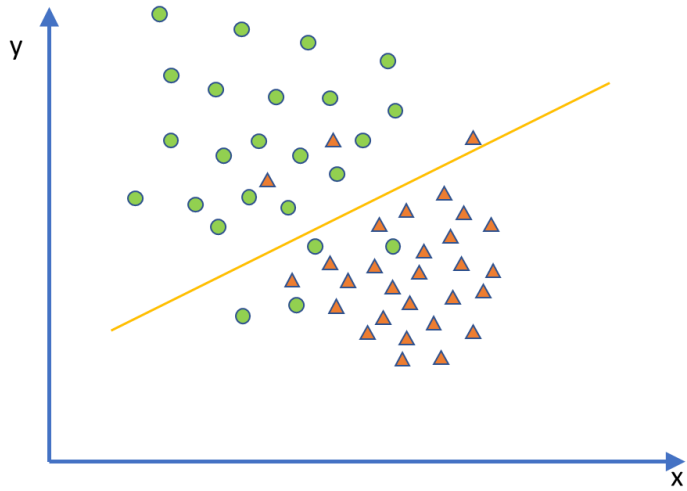
$$\mathbf{w}_{i+1} = \mathbf{w}_i + \eta \left(\frac{1}{N}\sum_{n=1}^{N}\frac{y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}_i^T \mathbf{x}_n}}\right)$$

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 found such that they maximize the likelihood that the process described by the model produced the data that were observed.

It is important to note that the training data set does not provide a probability of the class; rather it tells about 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, if any data point that lies above the orange color line is classified as class 1 and the data points which lies below the orange color line is 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 dependent variable is coded in 0 and 1. If the outcome is a success, than it is 1, or else it is 0.

For binomial logistic regression, the accuracy is measure using ROC-AUC, where ROC stands for Receiver Operating Characteristics.

Multinomial logistic regression deals with the situations where the dependent variable can have more than two outcomes. E.g., wine quality classification (class 1, class2, class3, etc.). it works on the concept of *one-vs-rest*. In one-vs-rest approach a binary model is learnt for each class, which tries to separate this class from all other classes, resulting in as many binomial models as there are classes. E.g., If the outcome can have 4 classes, then the multinomial logistic regression will create 4 binomial models to determine the classes. i.e. equation# 1 for class 1 or not class1, equation# 2 for class2 or not class2, equation# 3 for class3 or not class3 and equation# 4 for class 4 or not class4.

To make a prediction all these binomial models run and the classifier which 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 no 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 of between overfitting and generalization. Another important parameter is the penalty. Either of L1 or L2 penalty can be implemented to get better results. The L1 penalty is more useful for a model explanation as 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. No need to worry about violation of Ordinary Least Square assumptions
3. Does not require independent variables to be normally distributed
4. Prediction falls between 0 and 1

The downside of logistic regression:

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

Examples

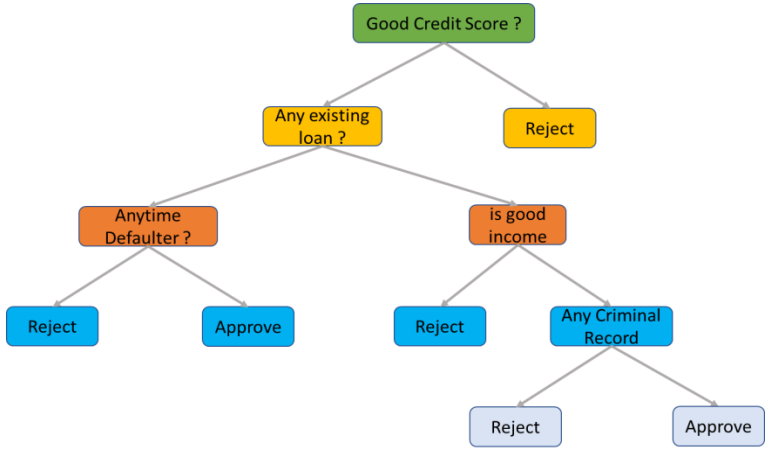
Logistic regression models are used for handwriting recognition, bank loan defaults prediction, bank loan approval prediction, credit card approval prediction, etc.

Decision Trees and Random forest

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

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

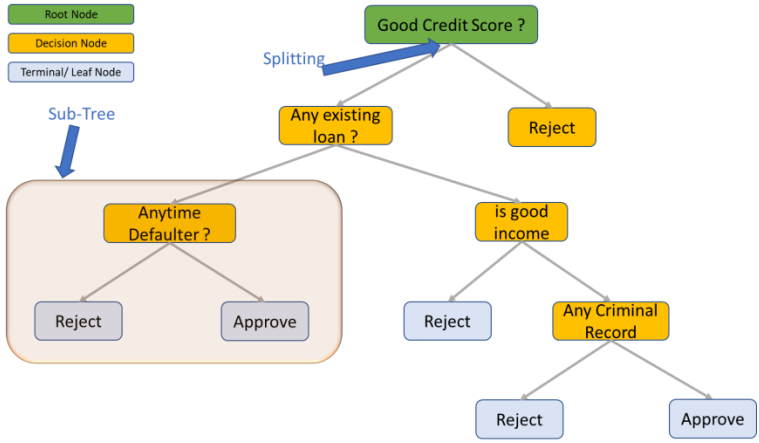
Consider a scenario where we want to decide whether a loan should be given to an applicant or not. To decide we will ask a series of questions to the applicant. We might start off with whether the applicant has any other existing loan. If the answer is yes, then the next question might be whether he 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 a model by hand to take a decision about whether a loan application should be approved or not. Instead of this, a machine learning model can be learnt from the data using supervised learning to take this decision. Decision tree model in machine learning can learn these decision conditions (test) from the data set quickly and build the model.

Decision tree models work best if the training data set 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 of a 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 as 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. Let's have a look at the figure below to visualize these notations.



The root node has the full data set, and at each decision node, the test is conducted to split the data set. 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 called pure.

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

Decision trees are prone to overfitting if the parameters are set to get all pure leaf nodes. Which means all data points in training dataset are correctly classified. To reduce overfitting commonly following strategies are 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 minimum no of points required to split it further.

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

Which feature to be used for splitting at each node is very important as the decision trees with a different split node may result in different prediction and accuracy. We can utilize a statistical method to identify the feature that should be selected as the root node. The feature which has 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's take an example:

> A training dataset has 500 observations. Out of these 500 observations, 300 are of positive class, and 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 has maximum information gain that will be selected for the split.

If we have a set of binary responses from some variable, all of which are positive/true/1, then knowing the values of the variable 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 ID3 are other popular methods to calculate information gain of a feature.

Decision trees are helpful in identifying what features are playing important role as well as to understand 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 hyper parameters of a decision tree model, e.g., the maximum depth of the tree, maximum no of leaf nodes, and minimum no of data point required to split the node further. Combination of these hyper parameters can be used to build the decision tree which is generalized over the data set and provides good accuracy.

Benefits of the Decision Tree:

1. Decision trees are helpful when a model explanation is required. Since it is built on various decision conditions, we can identify what all conditions it has

tested and what all have passed to reach to this decision.

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

The downside of Decision Trees:

1. Decision tress is prone to overfitting and provide poor generalization performance.
2. Low prediction accuracy.
3. It is a complex tree when there are many class tables.

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 way, and average their results to reduce overfitting.

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

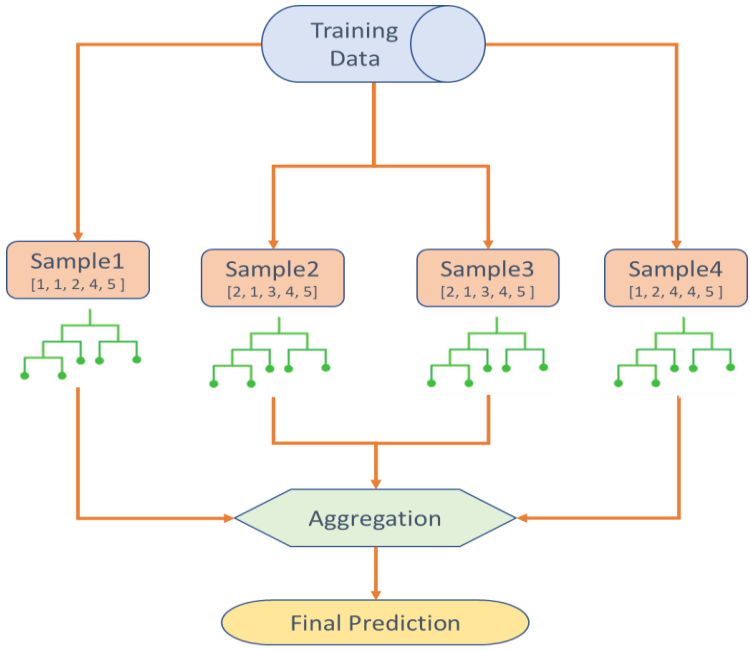
Bagging

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

Samples from the data set are drawn in a bootstrap manner, e.g. a sample of 10 observations is drawn from a training data set of 100 observations. These observations will be put back in the training data set before drawing another sample. So that next sample of 10 observations is drawn from the training data set of 100 observations. In simple term, at any point, all training data set 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.

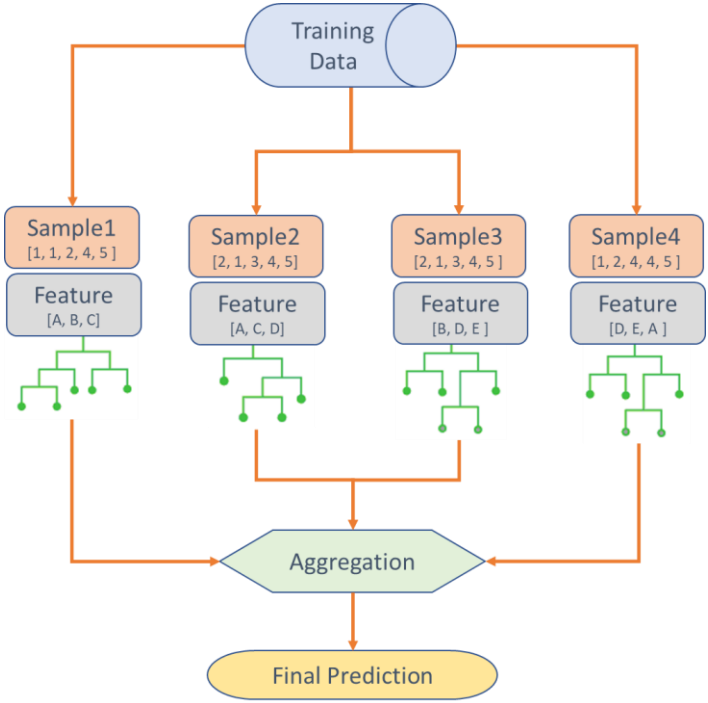
This is useful because models are created on various samples drawn from a single population. Bagging can reduce variance with little or no effect on bias.



Random Forest

Random forest models use bagging concept with slight modification. In bagging all features of the training set data are used on bootstrapped data (sample data) to create the base estimators. Since these sample data sets are almost similar, each base estimator mostly breaks at the same feature. This results in almost similar base estimators. And weak features importance will not be incorporated. To overcome this situation, 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 the features. This process of randomly selecting sample data and no 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.

Maximum no 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 $\log_2$(no of features) for regression use cases.

Criterion (***gini, entropy***), no of decision trees **(*n_estimators*)**, maximum no of features **(*max_features*)**, maximum depth of the decision tree **(*max_depth*)**, minimum no of samples required at each leaf node **(*min_samples_leaf*)**, minimum no of samples required to split a node **(*min_samples_split*)** and maximum no of leaf nodes in each decision tree **(*max_leaf_nodes*)** are 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.

Downsides of Random Forest:

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

Boosting

Contrary to 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 which are hard to predict. Boosting can reduce bias without incurring higher variance.

Popular boosting algorithms are Adaboost and Gradient Boosting.

Adaboost is adaptive learning where more attention is given to the records which are not correctly predicted. After each iteration weights of the wrongly predicted observations are increased so that these records will be picked more in the next iteration to get better accuracy.

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

Boosting does not introduce randomness to the decision tress however, but it uses strong pre-pruning techniques to build accurate predictors. In most cases, max depth for boosting models are kept to 5 only. 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.

No 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. Lower learning_rate means more trees are required and

higher learning_rate means less no of 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.

Downsides of Boosting:

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

Support vector machines

Support vector machines, commonly known as SVMs, are supervised learning algorithms. SVMs are mostly used for classification tasks. However SVM algorithm can be used for regression as well. SVM for classification is called as SVC (Support Vector Classifier), and for regression, it is called as SVR (Support Vector Regressor). SVMs can be used to build more complex models which are not simply defined by hyperplanes in input space.

Linear Support Vector Machines

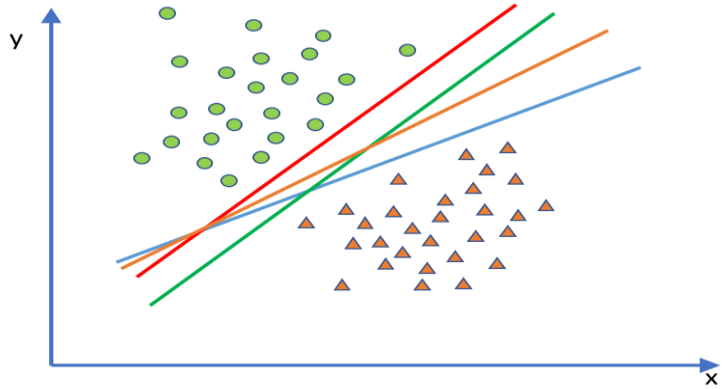
Linear models can do the classification using a line, plane or a hyperplane but these models do not perform well for non-linear datasets.

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 to 0.5, then the prediction confidence is low. SVM algorithm overcomes this situation by maximizing the distance between the classifier (line or hyperplane) and 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 margin. The result of SVM is the hyperplane for which margin is highest. Higher margin means there is more confidence in predicting the class accurately.

No of these support vectors is very small, and eventually, the hyperplane is identified from these support vectors only. If any of the support vectors are removed, it will alter the position of the hyperplane. The hyperplane is a linear

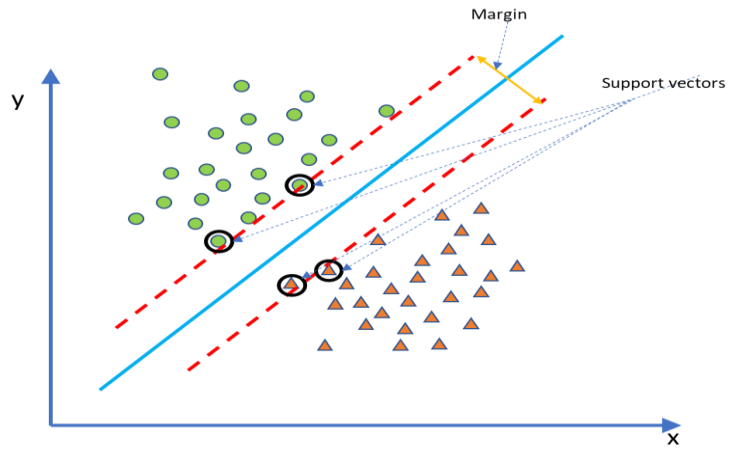
classifier build by an SVM algorithm. However, SVM algorithm can be extended to non-linear classification using kernel trick.

In this algorithm, each data item is plotted as a point in n-dimensional space, where n is the no of features in the dataset, and value of each feature of the observation is the value of a coordinate. Then we identify a hyperplane that can distinguish the classes accurately.



For a classification use, we can have multiple hyperplanes which can distinguish the data classes, but which one is best that can classify future data points accurately.

To get the best hyperplane SVMs to work on maximizing the distance between the nearest data point and the hyperplane. This distance is called the margin.



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

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, $f(x)$ 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 term is a loss. 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 generalized at the same time.

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

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

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

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

Including learning rate, the weight vector can be updated as below.

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

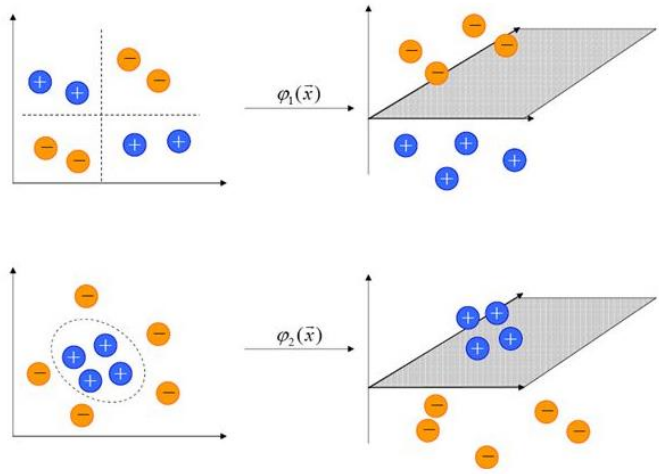
SVMs are great for relatively small datasets with fewer outliers. Decision trees and random forest can produce robust classifiers, but they require more data.

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

Until now, we talked about linear support vector machines where classes are separated by a hyperplane in a multidimensional coordinate system. These SVM algorithms can be extended to non-linear datasets using a kernel trick.

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



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 are 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.

SVM models work only for two class classification. However, in multi class scenario, it can create as many models as classes to compare one vs. rest. E.g., for class1 a model will be built to predict either class1 or not class1. Similarly, for class2 a model will be built to predict either class2 or not class2. For n classes, n models will be created.

Benefits of SVMs

1. SVM can create very complex decision boundaries for low dimensional datasets as well as for high dimensional datasets.
2. SVMs are effective for the cases where no of samples are less than no of features.
3. SVMs maximizes the distance between two classes hence provides more confidence level support to predicted values.

Downsides of SVMs

1. SVMs do not scale for huge no of samples. It needs a lot of memory and computation for large dataset.
2. SVMs expect all features to be on a similar scale.
3. Careful tuning of the hyperparameters is required.
4. SVM models are hard to explain

k Nearest Neighbors

The 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 are retained 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.

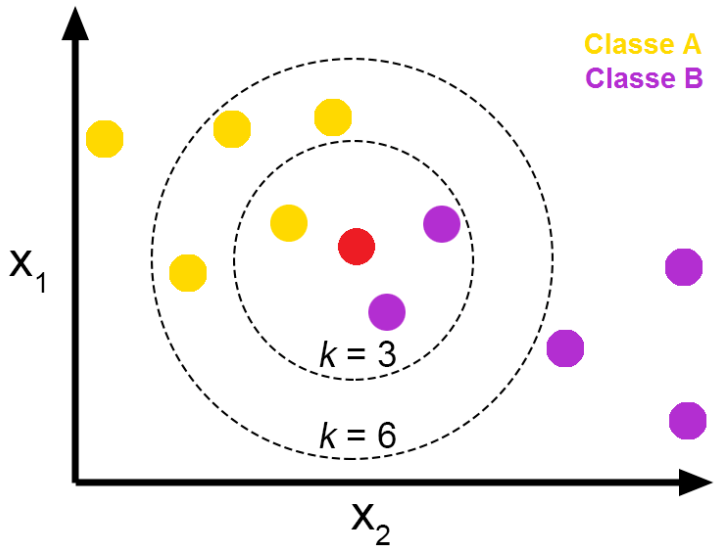
It is a lazy learning algorithm because the calculation is delayed until a prediction is required. This is called a localized model because only the data points which are near to new data point are used for model calculation, and to predict class for a new data point.

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 data points (nearest neighbors). 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 no of neighbors we want to check to classify a new data point. For k >1 the model uses voting to classify the new data point. In simple term, a class which is in the majority in k nearest neighbors is assigned to the new data point.

E.g., if the value of k is set to 3, means the model will check 3 nearest data points to classify the new data point. The default

value of k is 1, which means the vanilla KNN model classify the new data point as same as that of the class of the nearest neighbor.



In the above figure, yellow and purple color data points are from 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 nearest three 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 nearest 6 data points (outer circle) and then classify the red data point.

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

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

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

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

each class, and the class which 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. Default value of this parameter is 5.

- *Weights*: It holds a string value i.e., name of the weight function. The Weight function 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 used when all points in the neighborhood are weighted equally.

    - *distance* weight used for giving closer neighbors higher weight and far neighbors-less weight, i.e., weight points by the inverse of their distance.

    - *user defined function* we can call the user defined functions. The 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', brute'. It is an optional parameter.

- *ball tree , kd tree* are used to implement ball tree algorithm. These are special kind of data structures for space partitioning. *brute* is used to implement brute-force search algorithm.
- *auto* is used to give control to the system. By using 'auto', it automatically decides the best algorithm according to values of training data.fit()[xiii]

Benefits of k-Nearest Neighbor

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

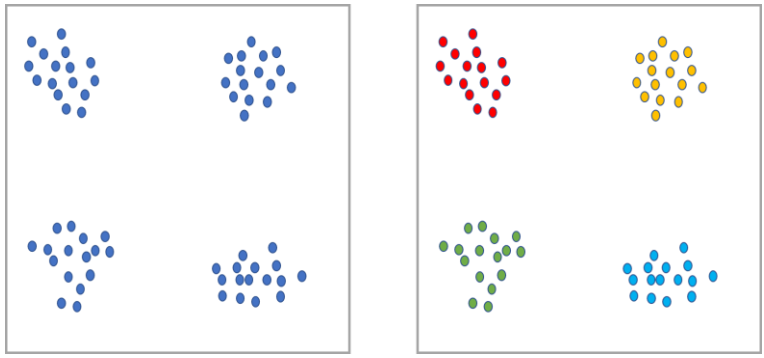
Downsides of k-Nearest Neighbor

1. Storage of data
2. For each prediction, calculation is done separately
3. Large search problems to find nearest neighbors

Clustering and K-means

Clustering is the most important unsupervised learning algorithm. It deals with finding a structure in a collection of unlabelled data. Clustering is defined as "the process of organizing objects into groups whose members are similar in some way".

A cluster is a collection of objects, which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. The figure below show a graphical representation:



In the example, four clusters can be identified into which the data can be divided; the similarity criterion is distance: two or more objects belong to the same cluster if they are "close" according to a given distance (in this example geometrical distance). This is called distance-based clustering.

Another kind of clustering is conceptual clustering: two or more objects belong to the same cluster if this one defines a

concept common to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

The goal of clustering is to determine the intrinsic grouping in a set of unlabelled data. Nevertheless, how to decide what constitutes a good clustering?

As many said, there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user, which must supply this criterion in such a way that the result of the clustering will suit their needs. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), or natural clusters, or unusual data objects (outliers detection), or useful data classes, or in describing their unknown properties ("natural" data types).[xiv]

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, earth quake 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 region of the data.

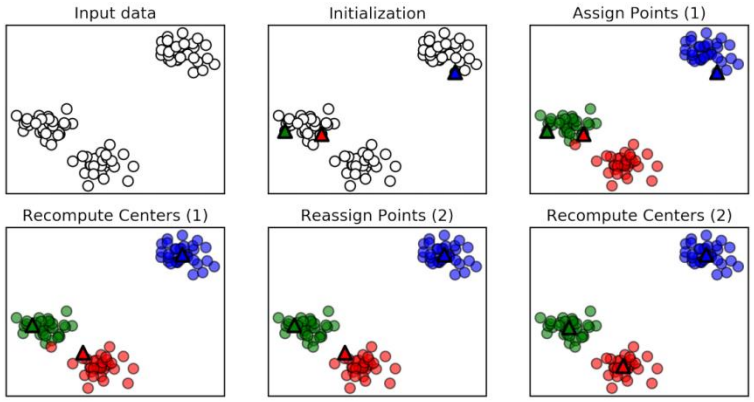
It is a two steps process:

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 mentioned steps until the assignment of data points to clusters no longer changes.

Once we decide how many clusters are required, the no of clusters is passed to the algorithm. Let's say for a given data set we want 3 clusters.



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

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

Assumptions in K-Means algorithm:

- All directions are equally important for each cluster.
- All clusters have the same diameter. it always draws the boundary between clusters to be exactly in the middle between the cluster centers.

Benefits of K-Means algorithm

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

Downsides of K-Means algorithm

1. Dealing with large number of dimensions and 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 doesn't 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. No of clusters should be known in advance.

6. 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 to 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 which are used for performance evaluation of a model.

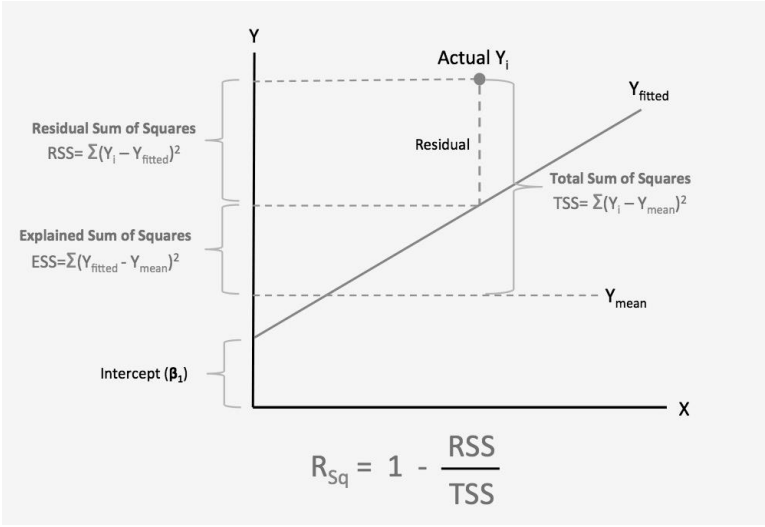
R-Squared (R^2)

Most common way to evaluate the performance of a linear model is by R-squared value. It is a statistical measure of determining how close is the data 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, 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 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 no of features/ predictors. Adding more no of features increases the R-squared. If no of features are too high, then the model starts predicting random noise in the data and tend to overfit.

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

Adjusted R-squared

To overcome the influenced behavior of R-squared on no of features, Adjusted R-squared is used. Adjusted R-squared is a modified version of R-squared that has been adjusted for the no of features in the model.

Adjusted R-squared penalizes the R-squared if the choice of the feature (newly added to model) isn't 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-square should be used to evaluate the model performance.

Confusion matrix

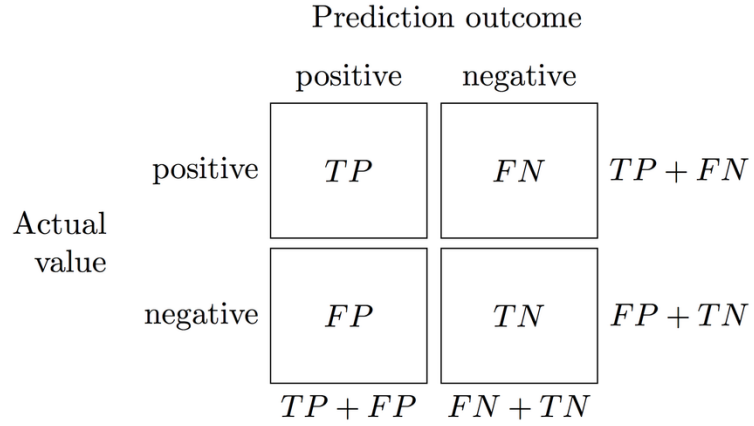
In classification scenarios, having good R-squared or Adjusted R-squared does not matter until both the classes are not identified with similar accuracy. This becomes a big issue when the data is not equally distributed for the target classes.

For example, consider that a dataset is having 98% observation of class A and 2% observations of class B, then 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 should be 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. 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 called 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 called as 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 called 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 – It indicates the proportion of records which are correctly classified. It is calculated as below.

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

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 the classes are correctly identified with the same accuracy or not.

E.g. total not of observation → 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 accuracy of identifying negative class is 0.

Recall or True positive rate (TPR) – It is a measure of how many actual positive classes are correctly identified. It is defined as the ratio of no of correctly predicted positive class to no of actual positive class. 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) – It is a measure of how many positive predicted classes are actually positive. It is defined as the ratio of no of correctly predict positive classes to no of predicted positive classes. High precision means incorrectly predicted positive classes (FP) are less. This metric is important when we want to avoid false positive.

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

f1-Score – Precision and Recall are important measures but looking at only one of them do not provide a clear picture of model performance. Most commonly used way of summarizing accuracy and recalls is the harmonic mean of these two. It is a better measure than accuracy as it takes precision and recall into account. This is also known as *f1*-score. It is calculated as

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

E.g. total not of observation → 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

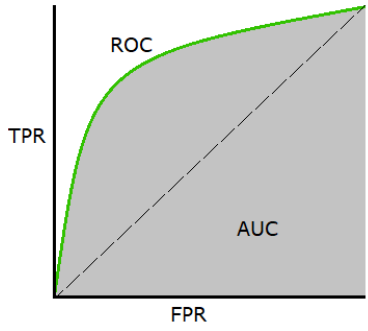
ROC Curve and AUC

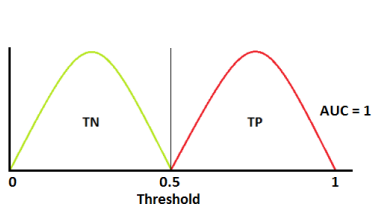
In classification, the default value of probability to classify into positive and negative classes is 0.5. This is also known as a threshold. E.g. p> 0.5 → class1, p< 0.5 → class0. The threshold value of 0.5 does not hold good for all the scenarios. We can change this threshold value to get the desired 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 threshold Receiver Operating Characteristics (ROC) curve is used. This is a plot between FPR (False positive rate) and TPR (True positive rate) for a given threshold.

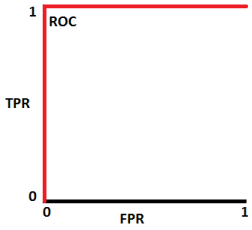
$$FPR = \frac{FP}{FP + TN} \quad TPR = \frac{TP}{TP + FN}$$

AUC (area under the curve) of RUC represents the degree of measure of separability. I.e., it tells how much the model is able to distinguish between classes. Higher AUC means, better the model is at predicting positive class as positive and negative class as negative.

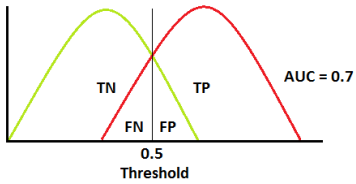
AUC= 1 means the model can clearly distinguish between the two classes.

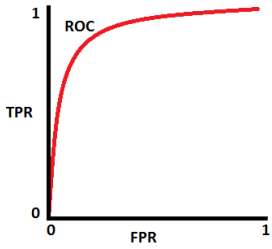




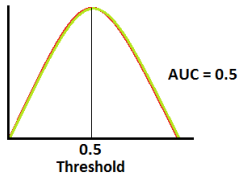


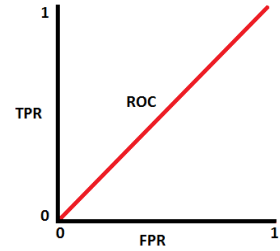
AUC = 0.7 means there are 70% chances that positive class will be identified as the positive and negative class will be identified as negative.



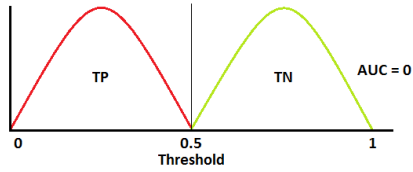


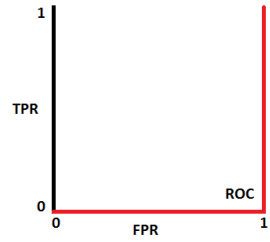
AUC = 0.5 means the model cannot clearly distinguish between two classes.





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



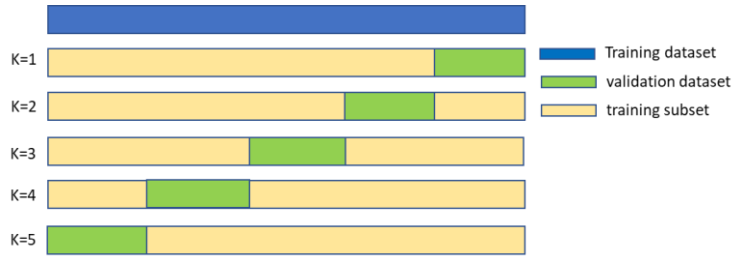


Cross-Validation

Cross validation is a tool that we use to utilize the training data set 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 make us more confident about the model trained on the training set. Without cross-validation, our model may perform well on the training set, but the performance decreases when applied to the testing set. The testing set is precious and should be only used once, so the solution is to separate one small part of the training set as a test of the trained model, which is the validation set.

K-fold cross-validation—In this, we split the training dataset into k subsets of data (also known as folds). ML 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 executed for all training subsets, average of error of each run is calculated and represented as cross validation error.

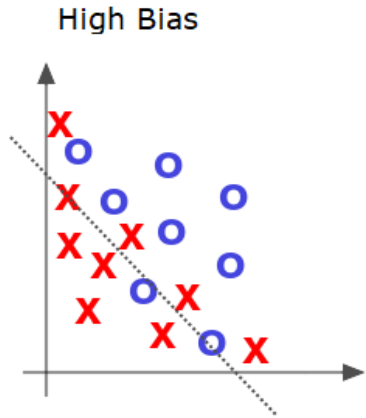


Leave-One-Out cross-validation—It is another technique used for cross-validation. It is logical extreme of K-fold cross validation where k = n (no of observations). Which means for each run only one observation is left a validation dataset. This approach leads to higher variation in testing model effectiveness because testing is done against one observation only. Hence the estimation gets 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: "Bias occurs when an algorithm has limited flexibility to learn the true signal from the dataset."[xv]

Bias is an algorithm's tendency to consistently learn the wrong thing by not taking into account all available information in the data. 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. E.g., if someone is biased towards something, then he is more likely to make wrong assumptions about them.



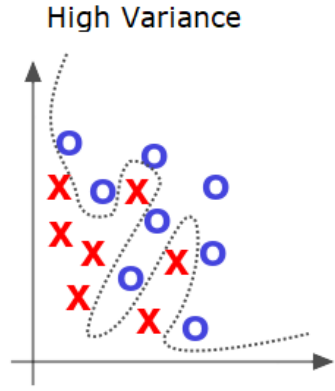
High bias can be reduced by adding more features, adding more polynomial features, or by decreasing regularization.

Variance

According to the EliteDataScience, variance refers to "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."[xvi]

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 data, rather than the intended output (overfitting). High variance makes the model to learn from noise, which means prediction will not be accurate.

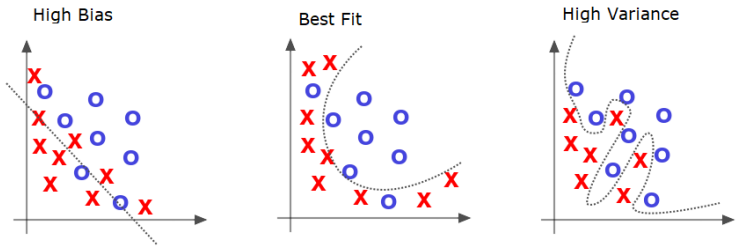


High variance can be reduced by collecting more training examples, using less no of features, or by increasing regularization.

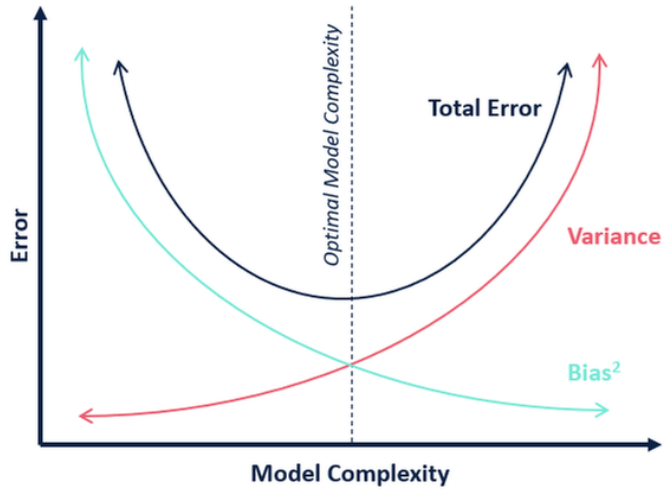
Bias-Variance tradeoff

High bias tends to highly generalize the model, and high variance tends to highly overfit the model. But we need an optimal model which is neither highly generalized nor highly overfit. To find this optimal model tradeoff needs to be done in between bias and variance.

With bias-variance tradeoff, we want to get the best model which is neither overfitted nor underfitted. It means the model should have low bias and low variance.



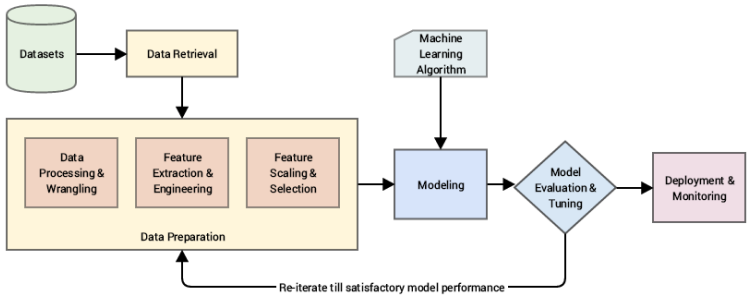
If the model is too simple, it results in underfit and complex model results in overfitting. Which means as the model complexity increases bias decrease and variance increases.



As shown in the above figure, we should create an optimal model which is not too complex and not too simple at the same time. This optimal model will have low variance and low bias.

Chapter 7: Best Practices

Many times, the dataset we get for machine learning problem 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 which can be utilized 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 model is trained. A dataset with good data and good quality features influence the model performance a lot. High level steps involved in a machine learning use case are depicted below.



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

Feature Engineering

Features are the independent variables and using these independent variables we predict the independent variable, also known as the target. Many times the independent variables available in the dataset have hidden information which the machine learning models cannot make the best use of it. To negate this situation in data preprocessing stage, we apply the domain knowledge and create new informative features out the existing features available in the dataset. These features should be created carefully; otherwise the model may tend to overfit. The set of features on which 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 which can make a huge difference between models.

Feature engineering is a recursive process, and broadly this process can be divided into the following steps:

1. Understanding dataset
2. Brainstorming 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, there is a use case to predict salary hike for employees of an organization. The dataset has employeeId, geographical location, date of birth, date of joining, date of starting the career, etc. In this dataset date of birth and date

of starting the career might not be useful for a dataset. If we derive two new features age of the employee and experience, then these two features could play a key role in the salary hike prediction. 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. Then how to transform the features which are non-numeric in nature. E.g., in employee dataset gender is maintained as a non-numeric feature, and this feature can have two values either Male or Female. For an organization 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 coding is one of the transformation technique that can be used to transform categorical features into numerical categorical features. If the feature has only two categories, then those two categories can be replaced by 0 and 1. In employee dataset example, gender feature we can replace Male with 0 and Female with 1.

Employee (Gender)
Male
Male
Female
Male
Female

One-hot encoding

Employee (is_female)
0
0
1
0
1

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

One-hot encoding can be extended to take care of multi category categorical features. It will create a new feature for each category and for each observation 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 employee rating example five new features will be created (is_NI, is_ME, is_S, is_EE, and is_EO).

Employee (last_rating)
NI
ME
S
EE
EO

One-hot encoding

is_NI	is_ME	is_S	is_EE	is_EO
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

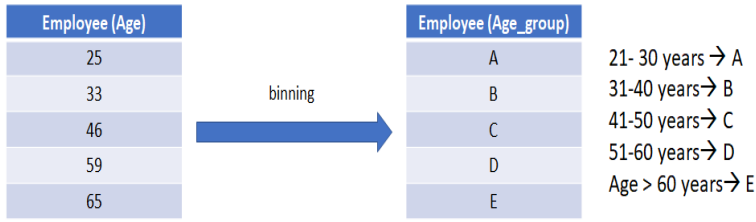
From these five derived columns, one column is obsolete. If four values are known, then the fifth value can be derived. E.g., in above example if is_NI column is removed then the value of is_NI column can be derived from the other four columns. Which means if all other four columns are 0, then it is 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. E.g. In employee example, we have a dataset of employees whose age varies from 21 to 60 years. These employees can be categorized in age brackets of 21-30, 31- 40, 41-50, 51-60, and above 60 years.

The technique of converting a continuous feature into multiples bins and creating a new feature out of it is knowns as binning or bucketizeation. It is also known as quantization. Binning transforms a continuous feature into a categorical feature, and categorical feature engineering might need to be done 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 no of bins/ buckets and 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 is equal to four then it is called as quartiles (divide data into 4 equal partitions).

3. Two-way Anova (Analysis of variance) test- it is used to find similarity between 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 employee example salary may vary from 40000 to 200000 and

age varies from 21 to 60. To improve the performance of these models, all features should be brought on one scale.

Commonly normalization (min-max scaling) and standardization techniques are used for feature scaling.

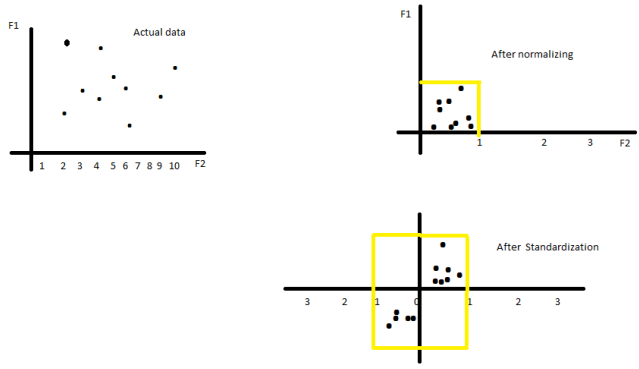
Normalization – In this feature scaling technique all features are transformed into 0 to 1 scale. Normalization results in smaller standard deviation which reduces the effect of outliers. It is achieved by subtracting the minimum value and dividing by the max value minus min value. It can be represented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

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

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

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

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



Data Imputation techniques

The dataset of many machine learning modeling use cases has missing values or outliers. One of the ways to handle it by discarding those observations, but this may result in small training dataset as well as 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 few common data imputation techniques. Data imputation for categorical and numerical features is different ways.

Numeric features data imputation:

1. Replacing with mean, median or mode – If a numerical 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, which means the feature has outliers and mean value is

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

2. Replacing with random sample value – to bring randomness in replacement of missing values as well, we select random observations from the data set and replace its feature value in the observation which has missing values. To introduce more randomness, we can select 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 in the dataset. E.g, the age of a person cannot reduce; hence this constraint should be kept in mind before estimating missing values for age.

Categorical features data imputation:

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

2. Replacing with mode – This technique is good to use when no of missing values is less. In this technique, the missing values are replaced by the category which occurs most in the feature.

3. KNN prediction – Apply KNN on other features to predict the feature which 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 to be imputed) can be grouped based on other feature(s). This controls biasness in the data imputation.

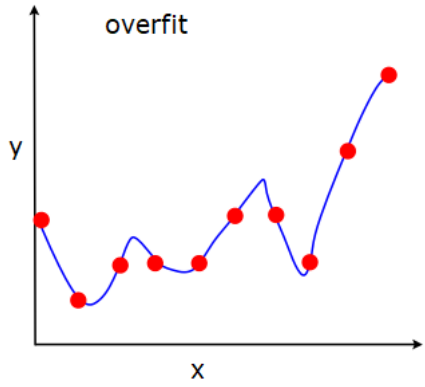
For missing values, it is always better to create a new feature to keep a track which observation has been updated by the data imputation technique. This new feature helps in reducing biasness introduced by the data imputation technique.

Overfitting and underfitting

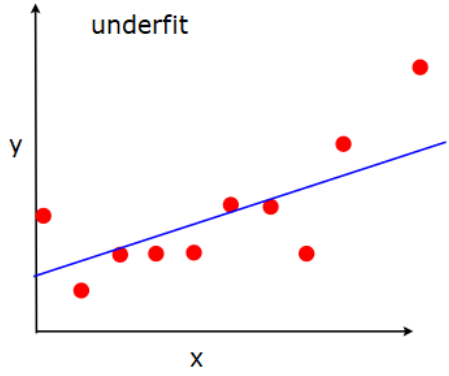
A typical machine learning model learns from the training data set and applies the learning on test dataset or unseen data. If the model is able to make correct predictions on unseen data, then the model is able to generalize the learning to unseen data.

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

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



Contrary to the overfit model, if the model is too simple and does not learn all the aspects and variations from the training dataset, then the model performs very poor on training data set, and it is called underfitted model.



The goal of modelling is to come up with a model which is not too much generalized and not too much focused on each individual data point. This model is called a best fit model.

This is basically a tradeoff between an underfit and an overfit model.

Regularization

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

Conclusion

Hopefully, this book has helped you to demystify the notion of machine learning. But that is just the beginning. Now that you've become familiar with the logic behind different types of learning, now that you've understood the role of statistics, and learned how to create simple algorithms, it is time to move on. But you won't be alone on that path. We have more books for you to reinforce your efforts at all stages of learning.

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've 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 of 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 **http://aisciences.net/books/** and you will find more books.

Next steps

Thanks you for purchasing this book. You now have a baseline understanding of the key concepts in Machine Learning.

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

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

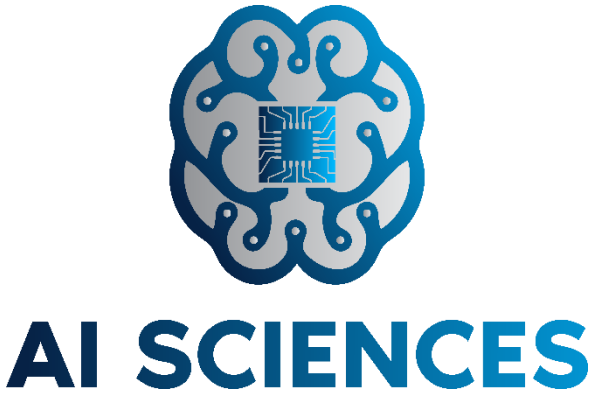
We highly recommend you to visit our website www.aisciences.net and subscribe to our email list. You will receive all our free books and you will be informed about all our promotions and offers.

Thank you !

Thank you for buying this book! It is intended to help you learning and mastering 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 problem, please let us know by sending us an email at review@aisciences.net before writing any 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] As above

[xiv] Clustering - Introduction, https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/ (accessed March 20, 2019).

[xv] Elite Data Science

[xvi] As above