



edunet
foundation

2023

Code Unnati

Powered by SAP

Implemented by EDUNET FOUNDATION



**This course booklet has been designed by Edunet
Foundation for the Code Unnati programme in
partnership with SAP**

Table of Contents

Learning Outcomes	9
Module I Machine Learning	10
Unit 1: Introduction Machine Learning.....	11
1.1 About Machine Learning	11
1.2 Machine Learning: Definition	12
1.3 How differ to traditional programming	13
1.4 Real time Applications of Machine Learning	15
1.5 Practical (Case Study): Stranger Things.....	20
1.6 Types of Machine Learning	21
1.7 Types of Data	23
1.8 ML Categories: Supervised & Unsupervised	24
1.9: Practical (ML HandsOn): GUI V/S Bare Coding.....	27
1.10 Coding Platform: Python.....	47
1.11 Anaconda: Introduction & Installation	49
1.12 Requisite libraries: Numpy, Pandas & Seaborn	61
1.13 Scikit-learn: Bring ML frameworks in Python	64
1.14 Flow Graph of ML Stages	65
1.15 Practical: Hello World to ML!.....	70
1.16 ML Problem Types.....	71
1.17 Need for Model Evaluation	73
1.18 About GitHub	74
1.19 Concept of Git.....	75
1.20 Git Staging	75
1.21 Git Commands.....	77
1.22 Practical: Working with GitHub Environment	79
Unit 2: Supervised Machine Learning	85
2.1 Linear Regression: Concept	85
2.2 Evaluation Metrics: Regression	93
2.3 Practical: Linear Regression using Python – Scikit Learn library	96
2.4 Bias-Variance Tradeoff	96
2.5 Handle Bias Variance Tradeoff: Lasso & Ridge Regression.....	98
2.6 Practical: Implementation of Ridge & Lasso Regression Using Python-Scikit Learn Library	99
2.7 Logistic Regression – Concept	99

2.8 Evaluation Metrics: Classification	102
2.9 Practical: Logistic Regression using Python-Scikit learn library.....	104
2.10 Cross Validation: Concept	104
2.11 K-Nearest Neighbors – Concept.....	105
2.12 Practical: K-NN using Python – Scikit Learn library	112
2.13 Decision Trees.....	113
2.14 Practical: Decision Tree Classifier using Sklearn.....	116
2.15 Random Forest: Concept.....	117
2.16 Bagging & Boosting in Machine Learning	120
2.17 Practical: Random Forest using Python- Scikit Learn library	123
2.18 Support Vector Machine: Concept.....	123
2.19 Practical: Support vector Machine Using Python	131
2.20 Ensemble learning	131
2.21 Practical: Ensemble Learning Using Python	139
Unit 3: Unsupervised Machine Learning	140
3.1 Unsupervised Learning	140
3.2 Introduction to Clustering	142
3.3 Various Distance Metrics	146
3.4 K Means Clustering	150
3.5 Practical: Implementation of K-Means Clustering using Python-Scikit Learn Library.....	153
3.6 The Silhouette method.....	153
3.7 Hierarchical Clustering Concept	156
3.8 Practical: Implementation of Hierarchical Clustering using Python- Scikit Learn Library.....	165
3.9 Dimensionality Reduction	165
3.10 Principal Component Analysis	169
3.11 Practical: Implementation of PCA Using Python- Sklearn Library.....	173
3.12 Linear Discriminant Analysis (LDA) in Machine Learning	173
3.13 Practical: LDA implementation using python	177
3.14 Applications and Drawback of LDA.....	178
3.15 Kernel Discriminant Analysis (Generalized Discriminant Analysis)	179
Unit 4: Machine Learning with IoT	181
4.1 Introduction: Machine Learning with IoT	181
4.2 Applications of ML with IoT	183

4.3 Practical: Transferring IOT Data to Cloud Services	185
4.4 Collecting Sensors Data from Cloud.....	188
4.5 Practical: Machine Learning on Sensor Data.....	189
Module II Internet of Things	190
Unit 1: Internet of Things.....	191
1.1 Internet Usage and Population Statistics	191
1.2 What is Internet of Things?.....	192
1.3 Why IoT?	193
1.4 IoT Architecture	195
1.5 What is industrial IoT?	196
1.6 IoT Applications by Industries.....	197
1.7 The Future of the Internet of Things	199
Unit 2: Sensors and Actuators.....	200
2.1 Sensors	200
2.2 Electronics Components.....	202
2.3 Electronics Signal	209
2.4 PWM – Pulse Width Modulation	211
2.5 ADC – Analog to Digital Converter	213
2.6 Types of sensors	216
2.7 Introduction to Actuators.....	223
Unit 3: IoT Protocol and Cloud integration	231
3.1 Introduction to Networking devices	231
3.2 Local and Personal Area Networks (LAN/PAN) for IOT	234
3.3 IOT WAN	235
3.4 IOT NODE	235
3.5 IOT Gateway.....	236
3.6 IPv4 vs IPv6.....	237
3.7 Multi-homing	239
3.8 IoT Protocol Stack	240
3.9 Types of Wireless Communication Protocols in IOT	244
Unit 4: IoT Hardware and Software & implementation.....	252
4.1 Introduction to Raspberry Pi	252
4.2 Install Raspbian OS in Raspberry Pi 4 B	255
4.3 Configure GrovePi+ Kit	257
Unit 5: Interface Grove Pi+ Sensors to Raspberry Pi.....	269

5.1 Practical: Interface Light Sensor with raspberry Pi	269
5.2 Practical: Interface Sound Sensor with raspberry Pi.....	272
5.3 Practical: Interface Grove LCD with Raspberry Pi	274
5.4 Practical: Interface Grove Button with Raspberry Pi.....	279
5.5 Practical: Interface Grove Temperature and humidity sensor (DHT22) with Raspberry Pi.....	281
5.6 Practical: Interface Grove Relay Switch with Raspberry Pi.....	283
5.7 Practical: Interface Grove Ultrasonic sensor with Raspberry Pi.....	286
5.8 Practical: Interface Grove Rotary Angle sensor with Raspberry Pi.....	289
5.9 Practical: Creating GUI interfaces to communicate with sensor and displays	291
Module III Deep Learning, Computer Vision & Edge Computing with OpenVINO toolkit	294
Unit 1: Deep Learning	295
1.1 What is Deep Learning?	295
1.2 Architecture.....	295
1.3 Deep learning vs. Machine learning.....	296
1.4 Concept of Neural Networks	298
1.5 Multilayer Perceptron	312
1.6 Gradient Descent.....	312
1.7 Loss Function	317
Unit 2: Computer Vision Basics	324
2.1 What is Computer Vision (CV)?	324
2.2 Image Fundamentals:	325
2.3 Computer Vision Using OpenCV	326
2.4 Practical: Canny Edge Detection	332
2.5 Convolutional Neural Network	334
2.6 Explanation of Convolutional Neural Network in detail	334
2.7 Transfer Learning	338
2.8 Transfer Learning Hands on	340
Unit 3: Computer Vision Hands-On	344
3.1 What is face detection?	344
3.2 What is Viola Jones algorithm?.....	344
3.3 Face Blurring in live video detection	347
3.4 Number plate detection.....	348
Unit 4: Computer Vision with OpenVINO	349

4.1 Introduction to OpenVINO	349
4.2 OpenVINO Toolkit Components	350
4.3 Model Optimizer.....	351
4.4 Benefits of OpenVINO	354
4.5 Practical: Installing Intel OpenVINO Toolkit (Linux)	355
4.6 Practical: Installing Intel OpenVINO Toolkit (Windows)	361
4.7 Exploring OpenVINO Toolkit Directories.....	364
4.8 Working with Model Optimizer	367
4.9 OpenVINO Deep Learning Workbench.....	370
4.10 Utilizing OpenVINO for inference at the edge	376
4.11 Edge Computing Using OpenVINO and Raspberry-Pi.....	387
4.12 Practical: Face Detection.....	388
4.13 Practical: Face, Age & Gender Detection using OpenVINO & R-PI.....	389
Module IV SAP Business Technology Platform ABAP Environment.....	391
Unit 1: Introduction to SAP Ecosystem.....	392
1.1 What is SAP	392
1.2 ERP Systems.....	393
1.3 Evolution of SAP	395
1.4 SAP Layered Architecture	399
1.5 Difference between SAP Functional & Technical Modules	401
Unit 2: Introduction to SAP Functional ERP Module.....	402
2.1 SAP PP (Production Planning)	402
2.2 SAP MM (Material Management).....	406
2.3 SAP PS (Project System)	410
2.4 SAP QM (Quality Management)	415
Unit 3: Introduction to SAP Technical ERP Module	419
3.1 SAP BASIS (Business Application Software Integrated Solution)	419
3.2 SAP Security.....	423
3.3 SAP HANA (High Performance Analytic Appliance)	426
3.4 SAP CRM (Customer Relationship Management)	436
3.5 SAP ABAP (Advanced Business Application Programming)	443
3.6 SAP BW (Business Warehouse).....	445
Unit 4: ABAP Language Basics & Class	450
4.1 Understanding the Basic feature of ABAP	450
4.2 Basic Data Objects & Data Types.....	451

4.3 Processing Data	454
4.4 Internal Tables	455
4.5 Defining a Local Class	457
4.6 Create Instance of a Class	458
4.7 Defining Methods.....	459
Unit 5: ABAP on SAP Business Technology Platform.....	461
5.1 Introduction to SAP Business Technology Platform.....	461
5.2 ABAP on SAP Business Technology Platform.....	463
5.3 Scenarios of ABAP on SAP BTP	464
Unit 6: Hands-On SAP Business Technology Platform ABAP Environment ..	467
6.1 Practical: Creating a BTP ABAP Environment.....	467
6.2 Practical: Creating an ABAP Package	474
6.3 Practical: Creating a Database Table	476
6.4 Practical: Create an ABAP Class.....	479
6.5 Practical: WAP to print Hello World	481
6.6 Practical: Perform Query Operation on the Output Table	482
6.7 Practical: Perform various ABAP data types of operation.....	483
6.8 Practical: Loops in ABAP, For loop, While Loop	486
6.9 Practical: Insert Data Table Entries and print them on the console	489
Reference	491

Learning Outcomes

After completing this handbook, learner will be able to

- Better Understanding of Machine Learning and its types
- Understand Significance of data and its sub-types
- Understand concept and working of major Supervised and Unsupervised Machine learning algorithms.
- Able to Implement various Machine Learning Algorithm in Python using Scikit Library
- Able to perform different types of dimensionality reduction techniques like PCA, LDA etc.
- Explain Concept of IoT and IoT Communication Protocol
- Able to interface various Sensors with Raspberry Pi
- Understand the concept and features of Deep Learning
- Able to solve problems using deep learning and neural network
- Understand concept of different layers in Convolution neural network
- Implement practical of computer vison using OpenCV
- Understand the concept and Workflow of OpenVINO Toolkit
- Model Zoo & Model Optimizers use for OpenVINO-IR
- Able to create application on Edge Computing Using OpenVINO & Raspberry-PI
- To get an idea about the SAP and its ERP Systems
- Understanding the layered architecture of SAP
- Getting insights of some of the functional ERP modules of SAP
- Introduction of different SAP Technical Modules, and understanding their basics Exploring the SAP Technical Module – ABAP basic features, Working with Data Objects and Classes
- Understanding ABAP on Business Technology Platform, Scenarios of ABAP on BTP
- Perform some hands-on practices with ABAP on SAP Business Technology Platform – Creating ABAP Environment, Creating Package, Create Class etc.

Module I

Machine Learning

Unit 1: Introduction Machine Learning

Learning Outcomes:

- Understand elements of Machine Learning and its types
- Understand significance of data and its sub-types
- Implement machine learning using GUI platform.
- Establish a bare coding environment for Machine Learning

1.1 About Machine Learning

Machine learning is behind chatbots and predictive text, language translation apps, the shows Netflix suggests to you, and how your social media feeds are presented. It powers autonomous vehicles and machines that can diagnose medical conditions based on images.

When companies today deploy artificial intelligence programs, they are most likely using machine learning — so much so that the terms are often used interchangeably, and sometimes ambiguously. Machine learning is a subfield of artificial intelligence that gives computers the ability to learn without explicitly being programmed.

“In just the last five or 10 years, machine learning has become a critical way, arguably the most important way, most parts of AI are done,” said MIT Sloan professor Thomas W. Malone, the founding director of the MIT Center for Collective Intelligence. “So that’s why some people use the terms AI and machine learning almost as synonymous ... most of the current advances in AI have involved machine learning.”

With the growing ubiquity of machine learning, everyone in business is likely to encounter it and will need some working knowledge about this field. A 2020 Deloitte survey found that 67% of companies are using machine learning, and 97% are using or planning to use it in the next year.

From manufacturing to retail and banking to bakeries, even legacy companies are using machine learning to unlock new value or boost efficiency. “Machine learning is changing, or will change, every industry, and leaders need to understand the basic principles, the potential, and the limitations,” said MIT computer science professor Aleksander Madry, director of the MIT Center for Deployable Machine Learning.

While not everyone needs to know the technical details, they should understand what the technology does and what it can and cannot do, Madry added. “I don’t think anyone can afford not to be aware of what’s happening.”

That includes being aware of the social, societal, and ethical implications of machine learning. “It’s important to engage and begin to understand these tools, and then think about how you’re going to use them well. We must use these [tools] for the good of everybody,” said Dr. Joan Larovere, MBA ’16, a pediatric cardiac intensive care physician and co-founder of the nonprofit The Virtue Foundation. “AI has so much potential to do good, and we need to really keep that in our lenses as we’re thinking about this. How do we use this to do good and better the world?”

1.2 Machine Learning: Definition

Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.

It is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behaviour. Artificial intelligence systems are used to perform complex tasks in a way that is like how humans solve problems.

In laymen’s terms, Machine Learning is about making predictions (answering questions) and classifications based on data. The more data you have, the easier it will be to recognize patterns and inferences. The data is used to train a machine learning model, then the model is used to make predictions and answer questions.

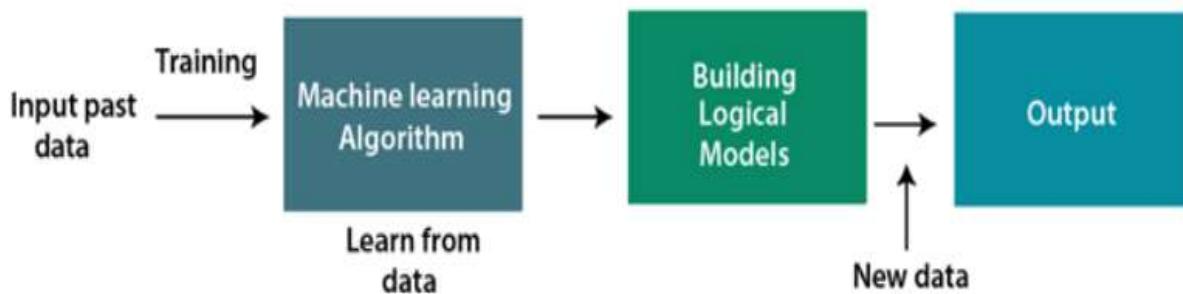


Fig: ML working concept

Reference: <https://static.javatpoint.com/tutorial/machine-learning/images/introduction-to-machine-learning2.png>

In some cases, writing a program for the machine to follow is time-consuming or impossible, such as training a computer to recognize pictures of different people. While humans can do this task easily, it's difficult to tell a computer how to do it. Machine

learning takes the approach of letting computers learn to program themselves through experience.

Machine learning starts with data — numbers, photos, or text, like bank transactions, pictures of people or even bakery items, repair records, time series data from sensors, or sales reports. The data is gathered and prepared to be used as training data, or the information the machine learning model will be trained on. The more data, the better the program.

1.2.1 Terminology of ML

Dataset: A set of data examples, that contain features important to solving the problem.

Features: Important pieces of data that help us understand a problem. These are fed in to a Machine Learning algorithm to help it learn.

Model: The representation (internal model) of a phenomenon that a Machine Learning algorithm has learnt. It learns this from the data it is shown during training. The model is the output you get after training an algorithm. For example, a decision tree algorithm would be trained and produce a decision tree model.

1.2.2 Process

Data Collection: Collect the data that the algorithm will learn from.

Data Preparation: Format and engineer the data into the optimal format, extracting important features and performing dimensionality reduction.

Training: Also known as the fitting stage, this is where the Machine Learning algorithm learns by showing it the data that has been collected and prepared.

Evaluation: Test the model to see how well it performs.

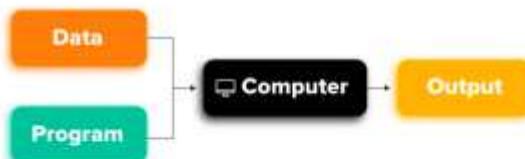
Tuning: Fine tune the model to maximize its performance.

1.3 How differ to traditional programming

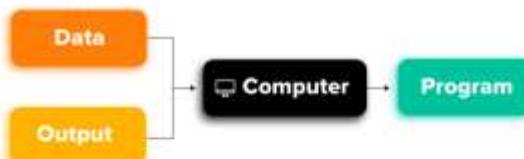
With digital becoming a part of everyday life, software development and programming continually keep revolutionizing businesses, especially across advanced intelligence. We are going to overview the differences between Traditional programming and Machine Learning paradigms.

In Traditional programming, we write down the exact steps required to solve the problem. While with a subset of Artificial Intelligence (AI), Machine Learning is motivated by human learning behaviour; we just show examples and let the machine figure out how to solve the problem by itself.

TRADITIONAL PROGRAMMING



MACHINE LEARNING



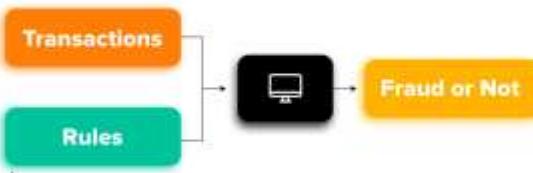
Reference: <https://www.avenga.com/wp-content/uploads/2021/12/image4-1.png>

The following is a real-life example of how traditional programming and Machine Learning differ. Imagine a hypothetical insurance company that is striving for the best customer experience in the 21st digital environment, as well as preserving assets and their ROI. So, the automatic detection of fraudulent claims is a part of their business processes.

Consequently, there are 2 possible scenarios of technology stepping in:

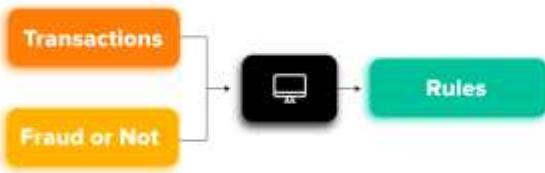
- Traditional programming (rule-based approach). In this case, we define the set of rules that will determine whether the claim is fraudulent or genuine, and then a developer translates them into code. This works great when such rules exist, and we know about them.
- Unfortunately, this is not always the case in real life. We do not always know exactly what rules a program should follow. The insurance agent's rule which reads: '*this case looks suspicious: deny the claim*' is difficult to translate into code, isn't it? Let alone the image classification problem, where it's almost impossible to come up with the rules that will allow us to differentiate between cats and dogs. And that's exactly where Machine Learning comes into play. The main property of Machine Learning algorithms is the ability to find rules using existing examples.

TRADITIONAL PROGRAMMING



Rule1. Claim time - Submit time < 1 h
 Rule2. Agreement review time > 5 m
 Rule3. ...

MACHINE LEARNING



Reference: <https://www.avenga.com/wp-content/uploads/2021/12/image3-1.png>

Now you may wonder what the actual algorithm behind training the model or 'learning' from examples is.

1.4 Real time Applications of Machine Learning

Machine learning is relevant in many fields, industries, and has the capability to grow over time. Here are six real-life examples of how machine learning is being used.



Fig: Applications of ML

Reference: <https://www.javatpoint.com/applications-of-machine-learning>

Let's try to pin down some great real-world applications of Machine Learning

1.4.1 Major Applications in Daily life

Google Maps' Traffic Prediction

One of the Machine Learning applications that we use in our day-to-day life, i.e., Google Maps' traffic prediction. Google Maps is very accurate in predicting traffic. If you have an Android phone or an iPhone with Google Maps opened and services enabled on it, then your phone or the app anonymously sends real-time data back to Google. Then, Google uses this information or data to calculate how many cars are there on the road or how fast they are moving. So, as more people start using the app, the traffic data becomes more accurate.

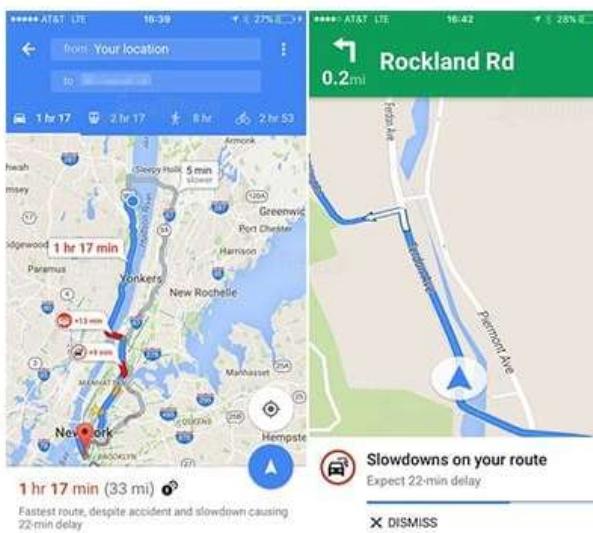


Fig: [Google-Map](#)

Reference: <https://www.businessinsider.com/how-google-maps-knows-about-traffic-2015-11?IR=T>

Google has also incorporated the traffic data from an app called 'Waze.' The company bought it for \$1 billion in 2013. The app monitors traffic reports from the local department of transportation. Google even keeps a history of traffic patterns on a specific road so that it can predict the traffic at a specific time on a specific location.

If traffic is more, the app would suggest a faster route to reach your destination on time. If you are feeling like Google is monitoring and tracking your every move, then you can opt-out anytime by turning off the location services. But what if everyone did that? Well, Google would be in trouble then as the data and the result might not be accurate. This is the reason why at some places Google Maps is not so accurate. So, this was about Google Maps using a Machine Learning algorithm to analyze and predict the result using your data. More data you feed more accurate it becomes!

Google Translate

The application of Machine Learning, Google Translate enables us to translate documents, sentences, and websites instantly. All these translations come from computers that use statistical machine translation. These translations are generated by computers based on the text patterns found.

For teaching someone a new language, usually start off by teaching the vocabulary, grammatical rules, and then explain about constructing sentences. Similarly, when a computer learns a new language, it understands that language by referring to the vocabulary and the grammatical rules. But learning a new language is very complicated because of the exceptions that exist in any rule. When you combine all of these exceptions in a computer program, then the quality of the translation begins to breakdown.

Now, when it comes to Google Translate, it takes a slightly different approach. Instead of teaching every rule of a language to the computer, what it does is, it lets the computer find the rules by itself. Google Translate does this with the help of Machine Learning. This is done by examining billions of documents that are already translated by human translators.

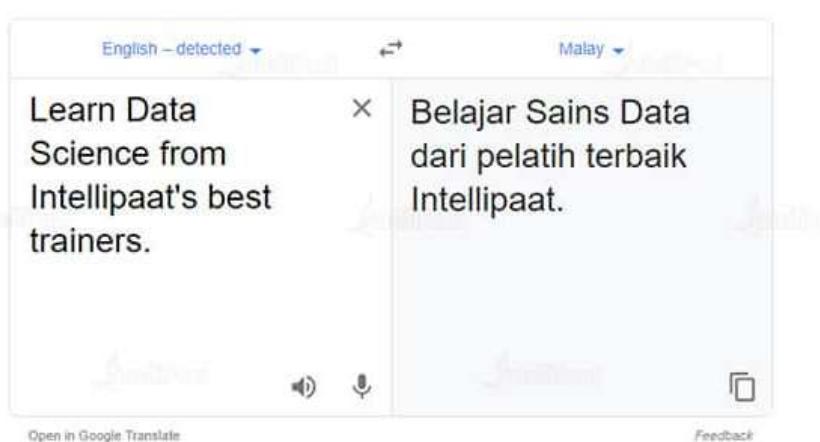


Fig: [Google-translate](#)

Reference: <https://blog.google/products/translate/24-new-languages/>

Facebook's Automatic Alt Text

Facebook's Automatic Alt Text is one of the wonderful applications of Machine Learning for the blind. Facebook has rolled out this new feature that lets the blind users explore the Internet. It is called Automatic Alternative Text. With the help of this, the blind are getting the tools by which they can experience the outside world and the Internet.

Fig: [Auto Alt Text](#)

Reference: <https://engineering.fb.com/2016/04/04/ios/under-the-hood-building-accessibility-tools-for-the-visually-impaired-on-facebook/>

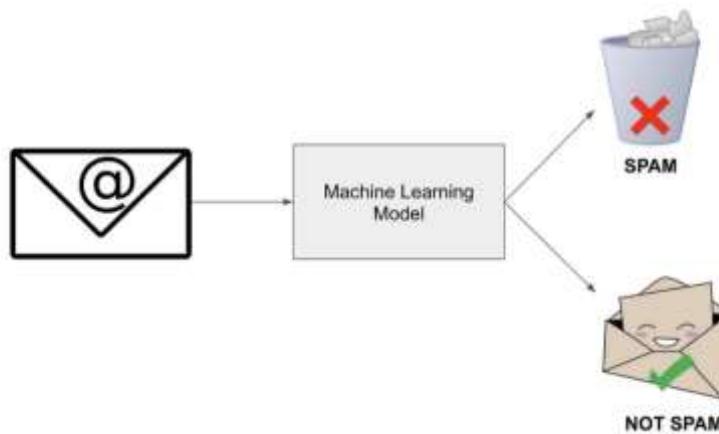
A normal person can see an entire screen full of information, and with that they can make decisions. But, for a blind person, it is a bit more difficult. They use screen readers that help in describing websites or apps. Also, they use keyboard shortcuts for jumping into a page. These screen readers usually look at the website code instead of what is visible on your screen.

Facebook has estimated that there are more than a billion photos shared every day. However, the pictures shared would be of no use for the blind if they don't come up with the text that outlines the picture.

So, Facebook is resolving this problem with the help of 'Automatic Alt Text.' Here, when the built-in reader is turned on and when we tap on a picture, then Facebook's Machine Learning algorithms try to recognize the features of the image and then create an alt text. This alt text will describe the picture with the help of the screen reader.

Spam Detection in Gmail

This is one of the common applications of Machine Learning that encounter in our day-to-day work. Spam detection makes use of filters. Algorithms are regularly updated based on the new potential threads found, advancement in technology, and the reaction given by users to spammed mails. Spam filters remove the threats using text filters based on the sender's history.

Fig:[SPAM Filtering](#)Reference: https://miro.medium.com/v2/resize:fit:1200/1*_igArwmR7Pj_Mu_KUGD1SQ.png

Amazon Alexa

The device shown in the below image is Amazon Echo, and the brain or voice of Echo is known as Amazon Alexa. It is capable of doing several tasks such as giving the weather report and playing your favorite song. Also, the word ‘Alexa’ is a wake word. As you say this word, it starts the recording of your voice. When you finish speaking, it sends the voice to Amazon. The service that persists this recording is called Alexa Voice Service or AVS, and it’s one of the magnificent applications of Machine Learning. This service is run by Amazon.

Fig: [Amazon-Alexa](#)Reference: <https://www.facebook.com/AmazonAlexa/>

1.5 Practical (Case Study): Stranger Things

Open below url

<https://evenstranger.pw/>



[Stranger things](https://evenstranger.pw/)

Upload any image on this website and AI will tell you the content of that image



1.6 Types of Machine Learning

Machine Learning is the go-to toolbox of the current business operations in a variety of domains. The implementation of machine learning into such operations is a strategic step and requires a lot of resources. Therefore, it is essential to understand what kind of business task you want your Machine Learning algorithm to work upon.

Based on the different flavors and objectives that a business can have, these machine learning algorithms are broadly classified as:

- Supervised Learning – “Teach me what to learn”
- Unsupervised Learning – “I will find what to learn”
- Reinforcement Learning – “I’ll learn from my mistakes at every step (Hit & Trial!)”

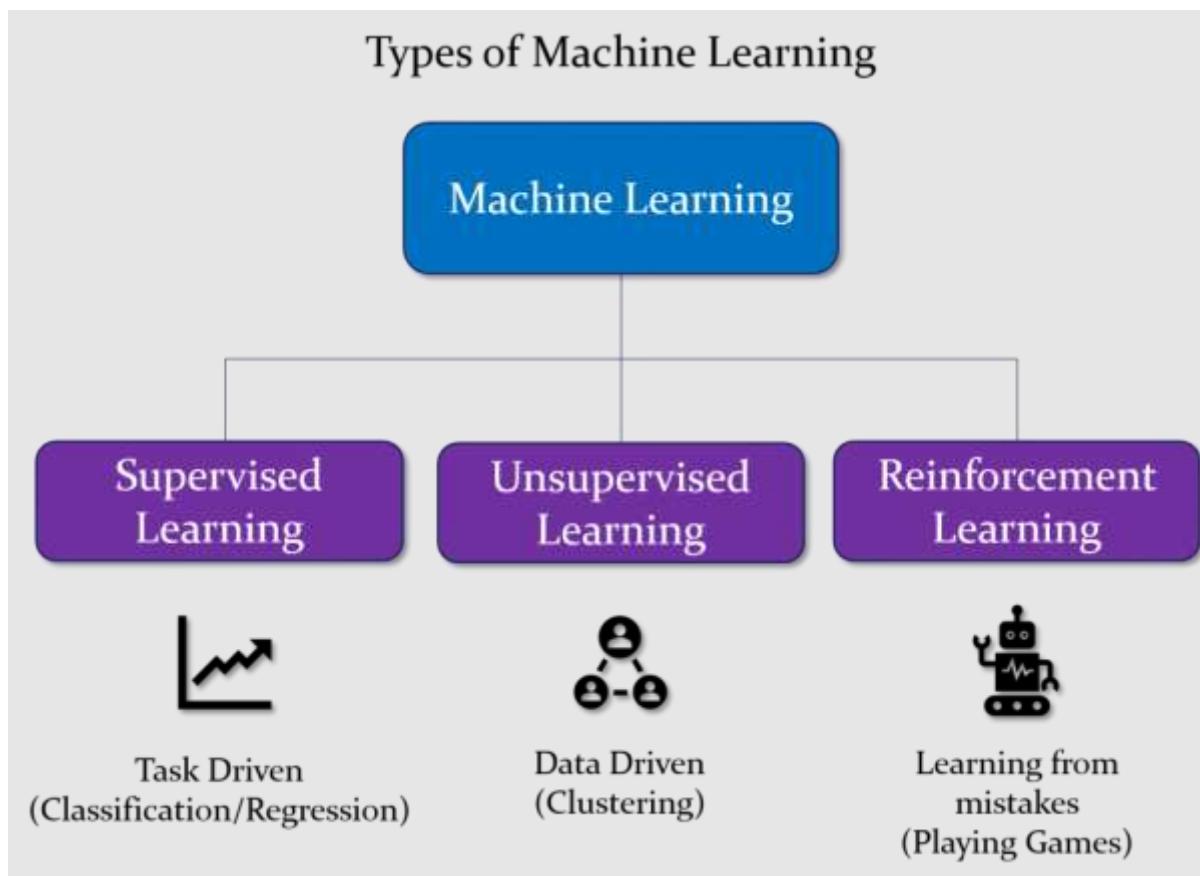


Fig: [Types of ML](#)

Reference: <https://www.newtechdojo.com/wp-content/uploads/2020/06/ML-Types-1-768x556.png>

The supervised learning: This approach is similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize (learn), and the student then derives general rules from these specific examples to use on a new example.

Unsupervised learning: Sometimes a kid does not need explicit supervision of a teacher. Instead, she tries to learn on its own based on her past experiences with the world. She memorizes (learns) what she has observed with the previous lessons (training data) and tries to replicate that in an unseen scenario (test data).

Reinforcement Learning: This form of learning is totally different from what we have discussed above. Sometimes, a kid might neither have any experiences nor any supervision to learn. But imagine that she has a godfather who gives her a candy whenever she learns something useful and punishes her when she makes a mistake. Alternatively, her training is now aided by rewards (candy) and punishments (corrections). This process makes her learn from her mistakes about where and how to improve. This is called as Reinforcement Learning.

1.7 Types of Data

There are different **types of data**, that are collected, analysed, interpreted and presented. The data are the individual pieces of factual information recorded, and it is used for the purpose of the analysis process. The two processes of data analysis are interpretation and presentation. Statistics are the result of data analysis. Data classification and data handling are important processes as it involves a multitude of tags and labels to define the data, its integrity and confidentiality.

The data is classified into majorly four categories:

- Nominal data
- Ordinal data
- Discrete data
- Continuous data

Further, we can classify these data as follows:

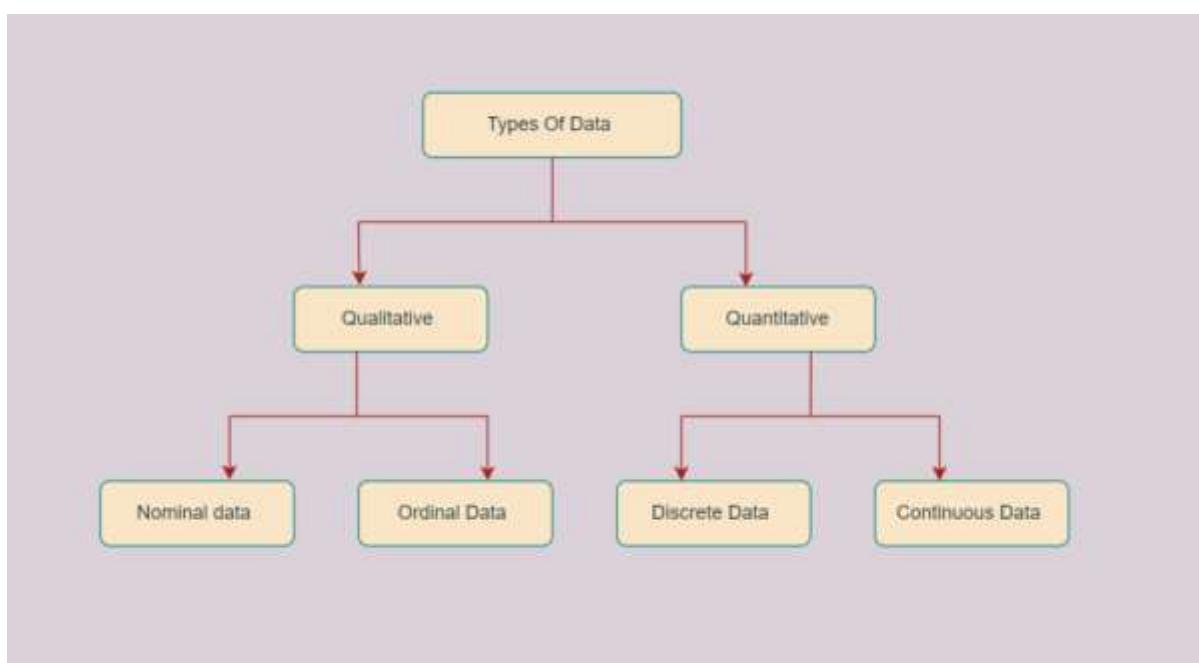


Fig:[Types of Data](#)

Reference: <https://d1m75rqqqidzqn.cloudfront.net/wp-data/2022/06/01113127/types-of-data-.png>

Qualitative or Categorical Data

Qualitative data, also known as the categorical data, describes the data that fits into the categories. Qualitative data are not numerical. The categorical information involves categorical variables that describe the features such as a person's gender, home town etc. Categorical measures are defined in terms of natural language specifications, but not in terms of numbers.

Sometimes categorical data can hold numerical values (quantitative value), but those values do not have a mathematical sense. Examples of the categorical data are birthdate, favourite sport, school postcode. Here, the birthdate and school postcode hold the quantitative value, but it does not give numerical meaning.

Nominal data: It is one of the types of qualitative information which helps to label the variables without providing the numerical value. Nominal data is also called the nominal scale. It cannot be ordered and measured.

Ordinal data: It is a type of data that follows a natural order. The significant feature of the nominal data is that the difference between the data values is not determined. This variable is mostly found in surveys, finance, economics, questionnaires, and so on.

Quantitative or Numerical Data

Quantitative data is also known as numerical data which represents the numerical value (i.e., how much, how often, how many). Numerical data gives information about the quantities of a specific thing. Some examples of numerical data are height, length, size, weight, and so on. The quantitative data can be classified into two different types based on the data sets. The two different classifications of numerical data are discrete data and continuous data.

Discrete data: It can take only discrete values. Discrete information contains only a finite number of possible values. Those values cannot be subdivided meaningfully. Here, things can be counted in whole numbers.

Example: Number of students in the class

Continuous data: It is data that can be calculated. It has an infinite number of probable values that can be selected within a given specific range.

Example: Temperature range

1.8 ML Categories: Supervised & Unsupervised

The two main categories of machine learning techniques are supervised learning and unsupervised learning.

1.8.1 Supervised Machine Learning

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output. In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

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

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

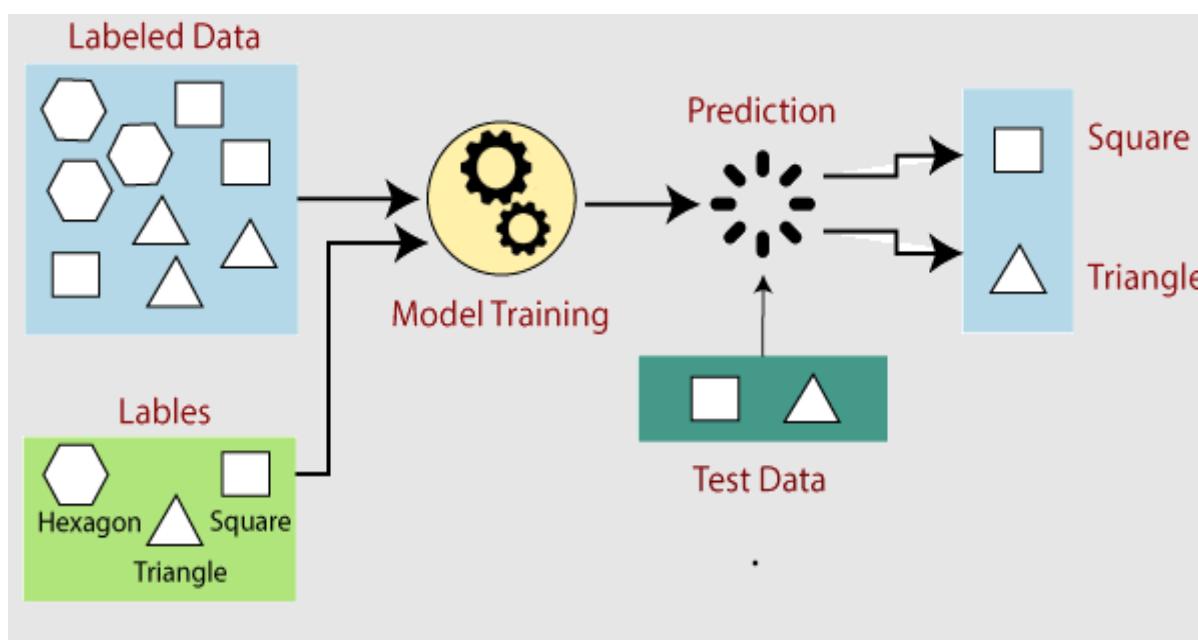
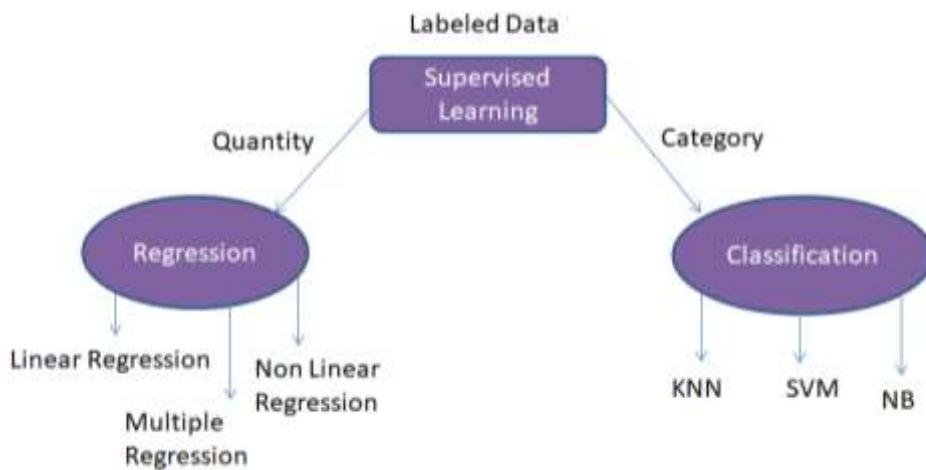


Fig: Working of Supervised learning

Reference: <https://www.javatpoint.com/supervised-machine-learning>

Types of supervised Machine learning Algorithms Supervised learning can be further divided into two types of problems:

Fig: [Categories of supervised learning](#)

Reference: <https://cdn.educba.com/academy/wp-content/uploads/2019/12/TYPE-OF-SUPERVISED-LEARNING-ALGORITHM.jpg>

1.8.2 Unsupervised Machine Learning

Unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things.

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

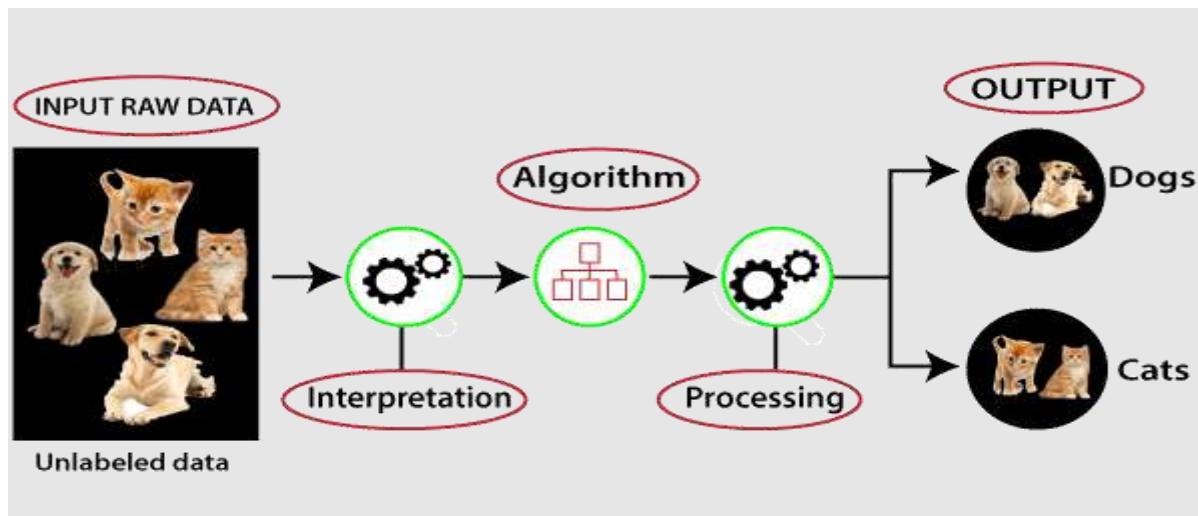
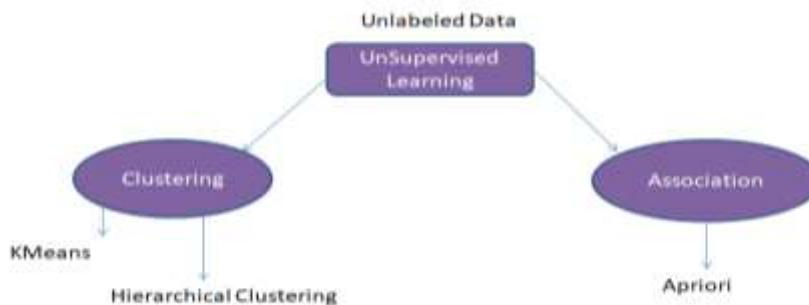


Fig: Working of unsupervised learning

Reference: <https://www.javatpoint.com/unsupervised-machine-learning>

Types of Unsupervised Learning Algorithm:

The unsupervised learning algorithm can be further categorized into two types of problems:

Fig:[Categories of unsupervised learning](#)Reference: <https://www.javatpoint.com/unsupervised-machine-learning>

1.9: Practical (ML HandsOn): GUI V/S Bare Coding

Azure Machine Learning Studio is web-based integrated development environment (IDE) for developing data experiments. It is closely knit with the rest of Azure's cloud services and that simplifies development and deployment of machine learning models and services.

Creating the Experiment

There are five basic steps to creating a machine learning example.

Obtaining The Data

Gathering data is one of the most important step in this process. Relevance and clarity of the data are the basis for creating good prediction models. Azure Machine Learning Studio provides a number of sample data sets.

- After collecting the data, we need to upload it to the Studio through their simple data upload mechanism:
- Our next step is to create a new experiment by dragging and dropping modules from the panel on the left into the working area.

Preprocessing Data

Preprocessing available data involves adjusting the available data to your needs. The first module that we will use here is “Descriptive Statistics”. It computes statistical data from the available data. Besides “Descriptive Statistics” module, one of the commonly used modules is “Clean Missing Data”. The aim of this step is to give meaning to missing (null) values by replacing it with some other value or by removing them entirely.

Defining Features

This module determines the features of the dataset that are most relevant to the results that we want to predict.

Choosing And Applying An Algorithm

Next step is to split the available data using the “Split” module. The first part of the data will be used to train the model and the rest is used to score the trained model.

- The following steps are the most important steps in the entire Azure machine learning process. The module “Train Model” accepts two input parameters. First is the raw training data, and the other is the learning algorithm.
- Evaluate Model module gives us an evaluation of the trained model expressed in statistical values.

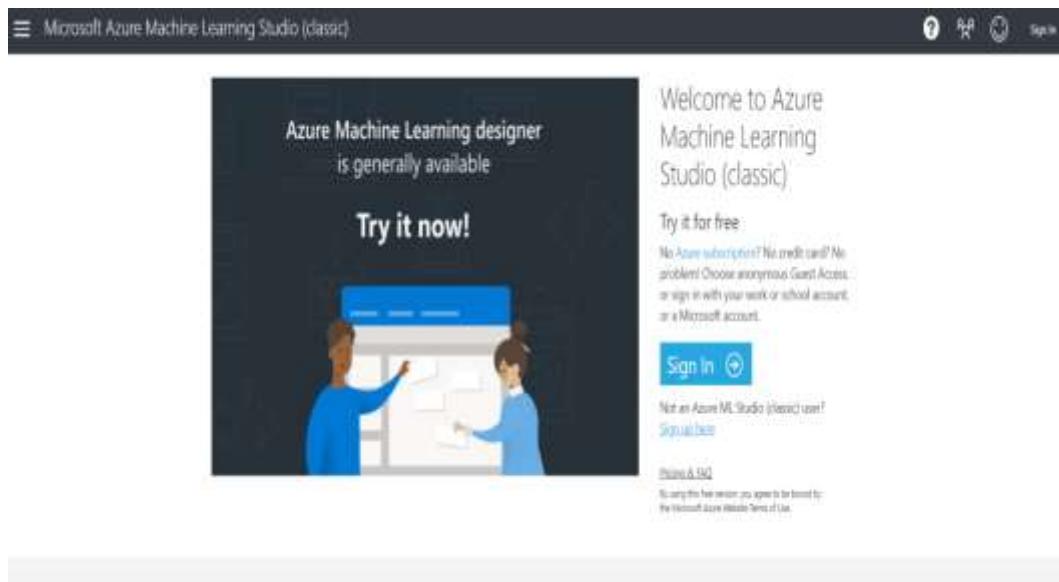
Predicting New Data

Finally, we can test our prediction web service using a simple test form.

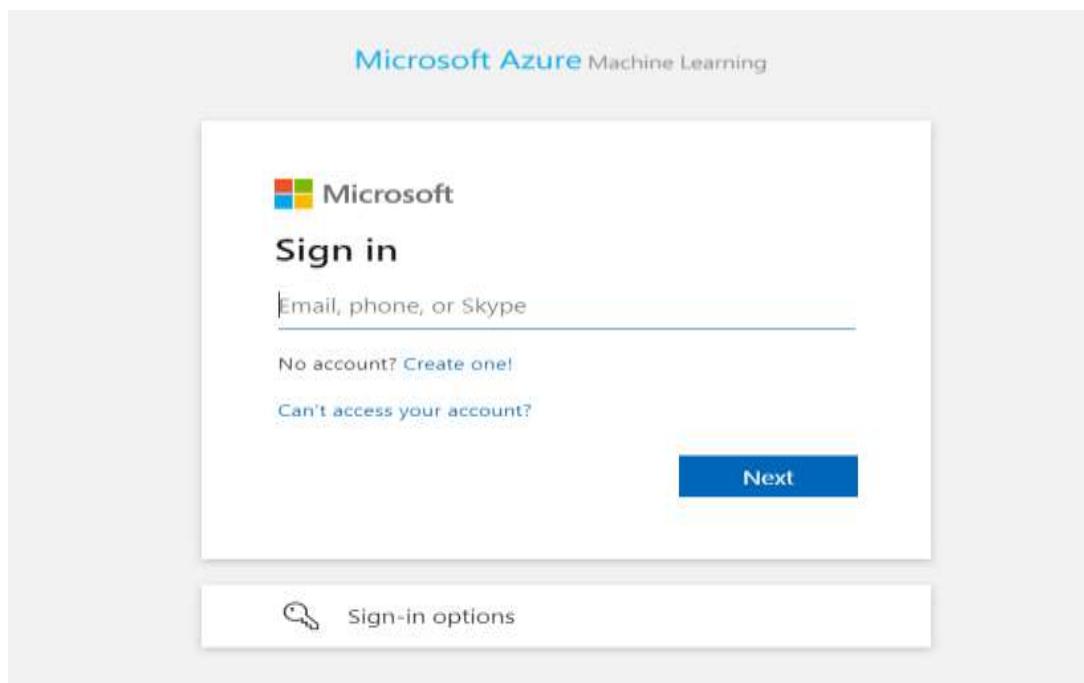
Setting up an account in Azure ML Studio and Creating Workspace

- Open the following web link on the web browser.

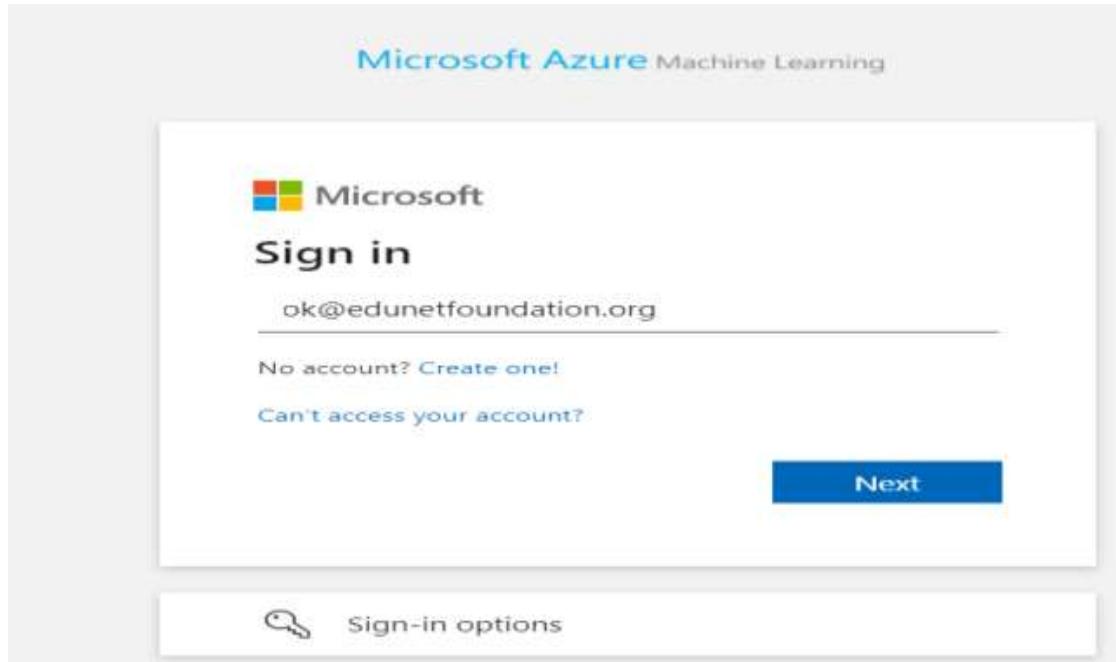
<https://studio.azureml.net/>



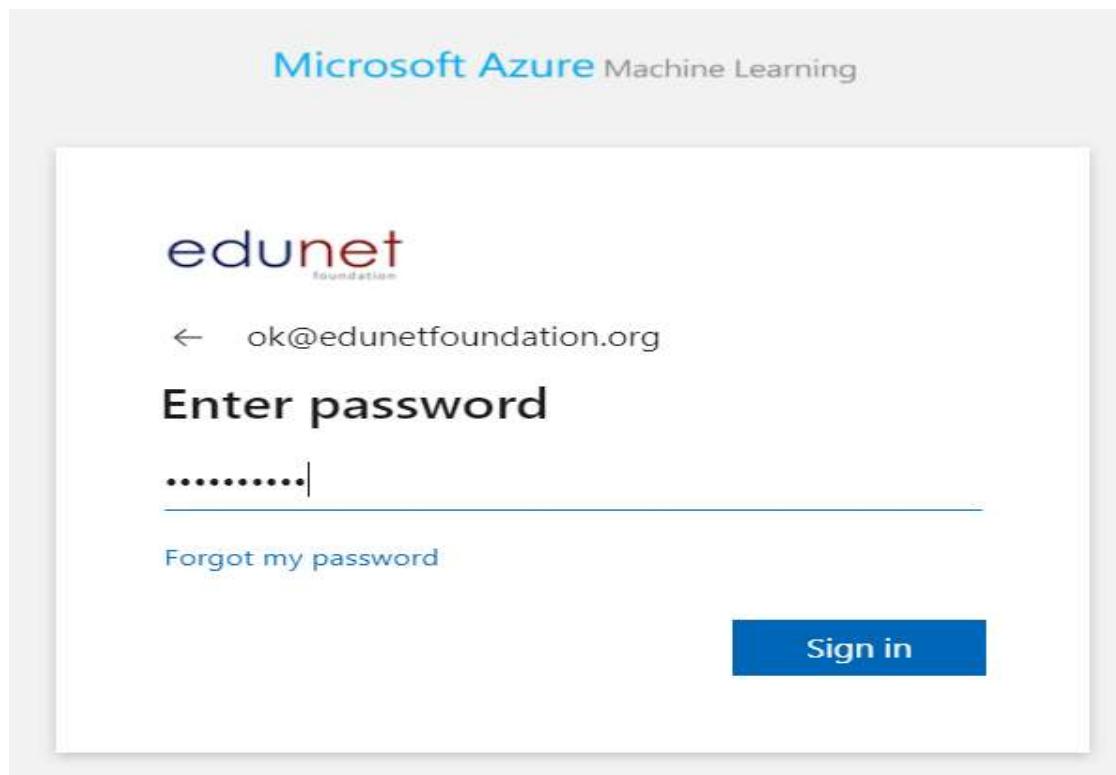
- Click on the Sign In.



- Type your Microsoft email ID and then press next. If you do not have the Microsoft account then create first using **create one** option (<https://signup.live.com/>).



- Type the password for your Microsoft account and press sign in.



- Press Yes to stay sign in.

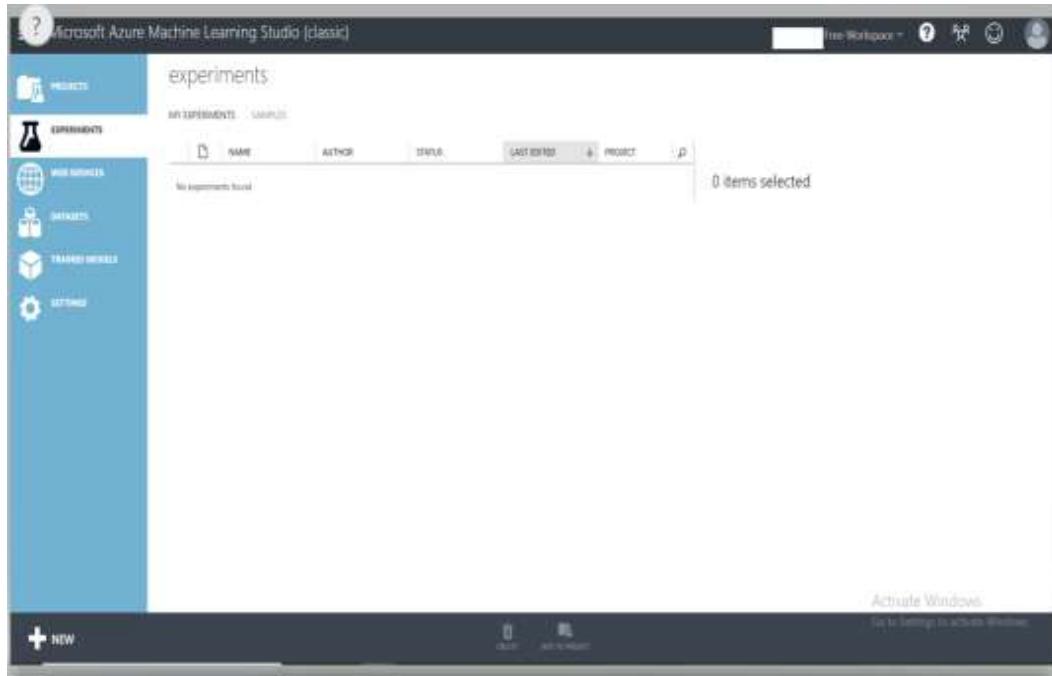


- Now, You will be redirected into following Microsoft Azure Machine Learning Studio (Classic) and your free workspace will be created as below:



Training a ML model in Azure Studio

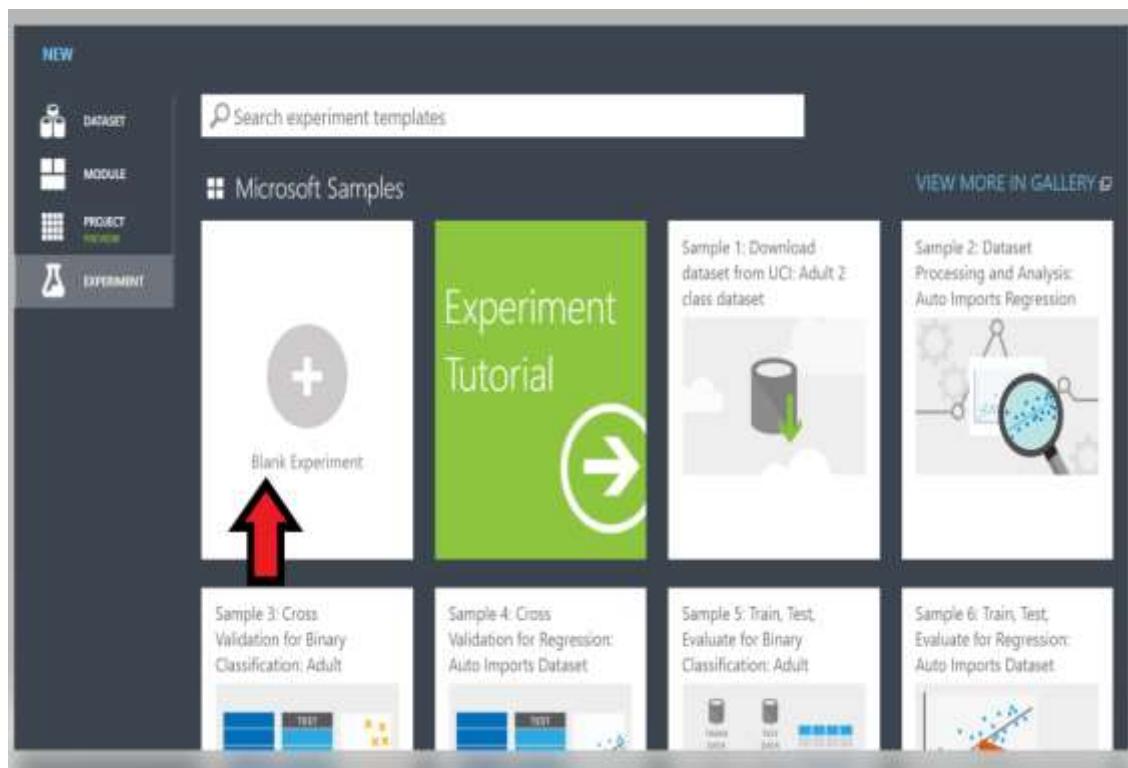
- Sign in into Microsoft Azure Machine Learning Studio (classic) and create workspace.



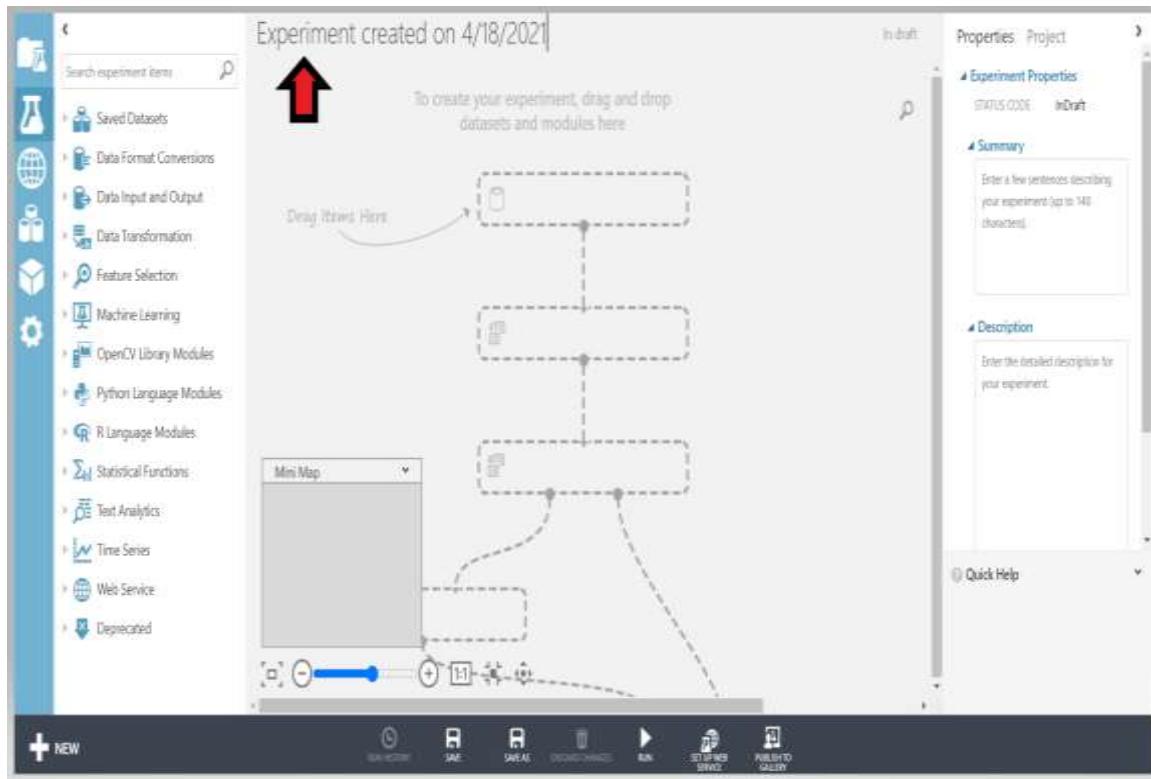
- First select **Experiment** and then **New** at the bottom of the page.



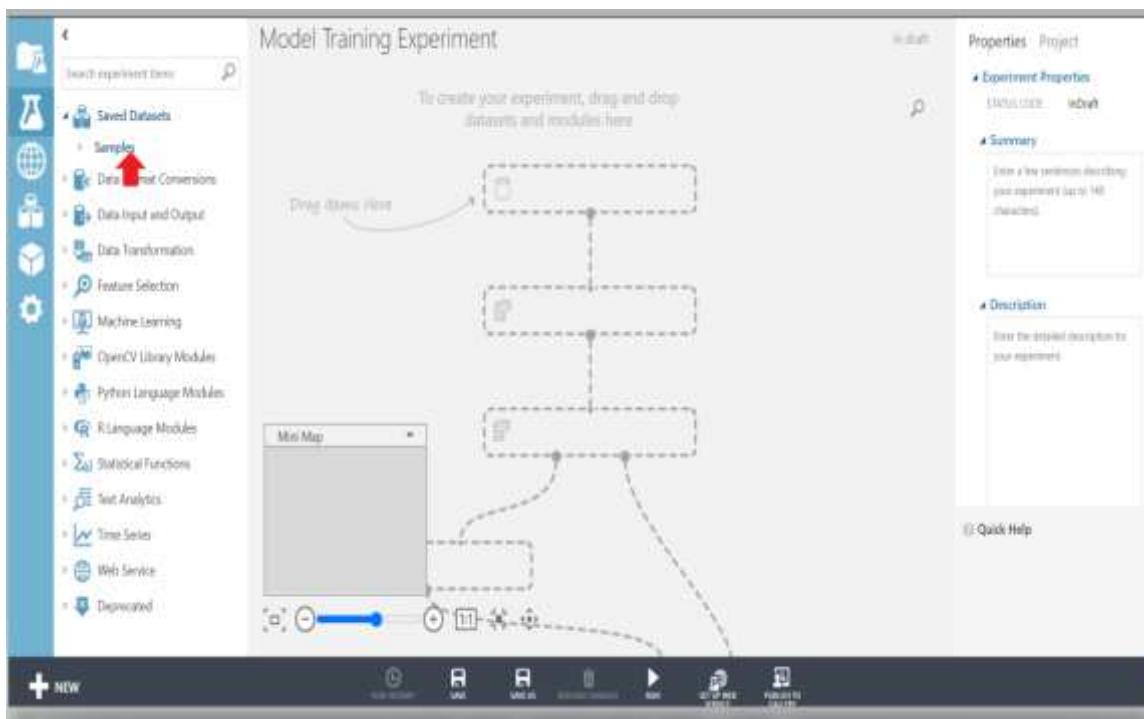
- Select Blank Experiment.



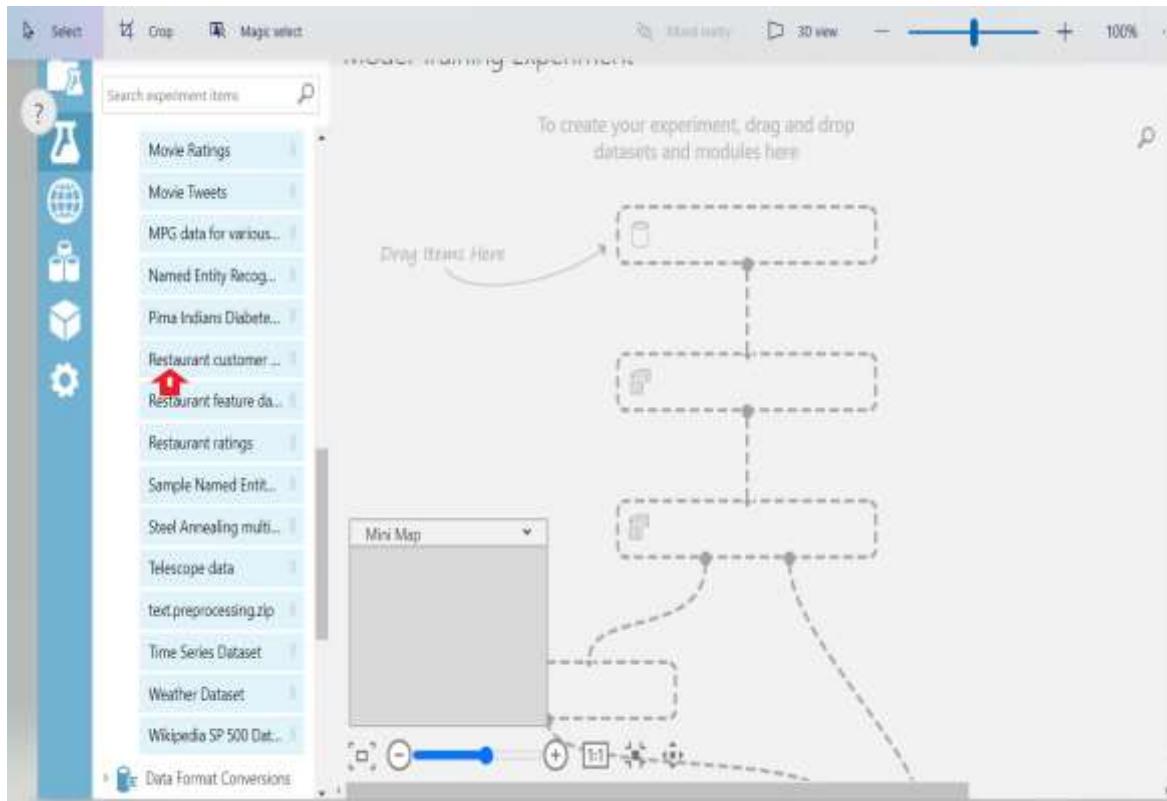
- Give the title for the project.



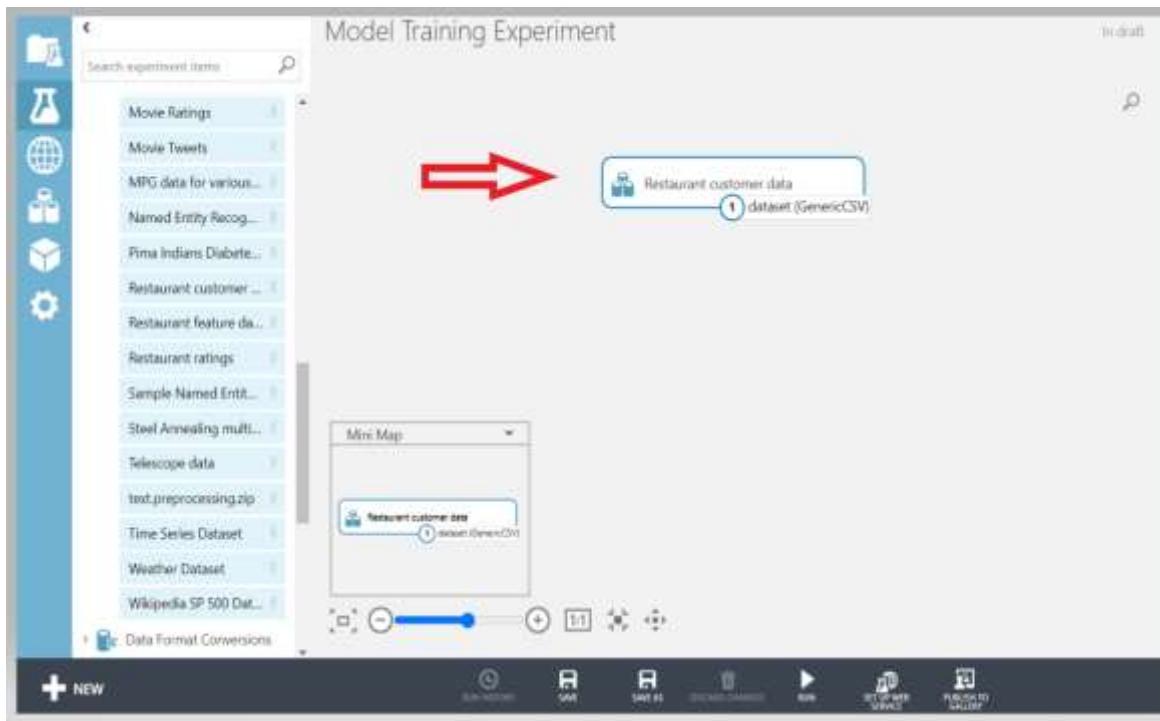
- Select **Sample** option from the **Saved Dataset**.



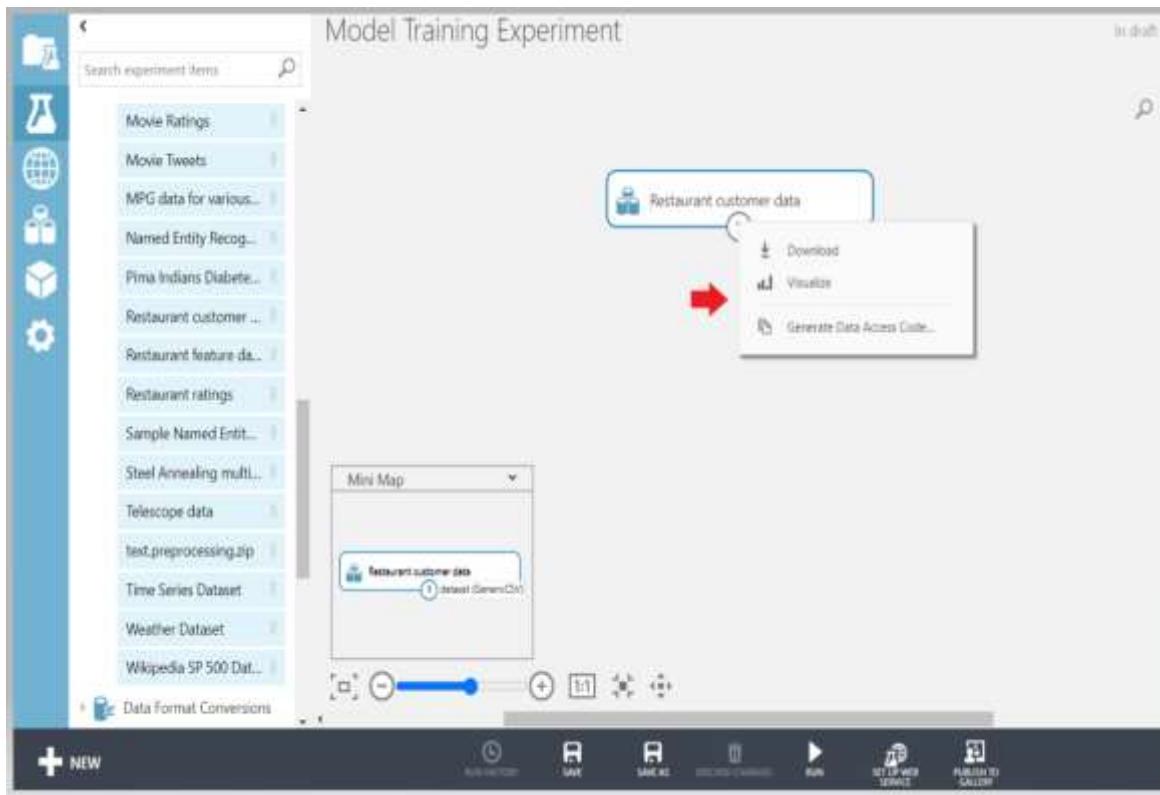
- Select Restaurant Customer Rating Dataset



- Drag selected dataset on Panel.



- Right click on 1 and choose visualize option.



- Visualize the Dataset and then close it:

Model Training Experiment > Restaurant customer data > dataset

	rows	columns						
	138	19						
view as:	grid	list						
	userID	latitude	longitude	smoker	drink_level	dress_preference	ambience	transport
	U1001	22.139997	-100.978803	false	abstentious	informal	family	on foot
	U1002	22.150087	-100.983325	false	abstentious	informal	family	public
	U1003	22.119847	-100.946527	false	social drinker	formal	family	public
	U1004	18.867	-99.183	false	abstentious	informal	family	public
	U1005	22.183477	-100.959891	false	abstentious	no preference	family	public
	U1006	22.15	-100.983	true	social drinker	no preference	friends	car owner
	U1007	22.118464	-100.938256	false	casual drinker	informal	solitary	public
	U1008	22.122989	-100.923811	false	social drinker	formal	solitary	public

To view, select a column in the table.

- Select the Data Transformation then switch to Manipulation then switch to Select Columns in Dataset and drag it into Panel.

Model Training Experiment

In Draft

Properties Project

Experiment Properties STATUS CODE In Draft

Summary Enter a few sentences describing your experiment up to 140 characters.

Description Enter the detailed description for your experiment.

Quick Help

Search experiment items

Data Format Conversions

Data Input and Output

Data Transformation (highlighted with a red arrow)

Filter

Learning with Counts

Manipulation (highlighted with a red arrow)

Add Columns (highlighted with a red arrow)

Add Rows

Apply SQL Transform...

Clean Missing Data

Convert to Indicator...

Edit Metadata

Group Categorical Va...

Join Data

Remove Duplicate Ro...

Select Columns in Da... (highlighted with a red arrow)

Select Columns Trans...

Restaurant customer data

Select Columns in Dataset

Mini Map

Restaurant customer data

Select Columns in Dataset

NEW

- Make connection between dragged item, press on red sign and then Launch column selector to choose the relevant column.



- Select all the relevant column by using **Ctrl Key** then use **>** option to move them right.

Select columns

The screenshot shows a "Select columns" dialog box. On the left, there are two sections: "BY NAME" and "WITH RULES". The "WITH RULES" section has an arrow pointing to the "smoker" column in the "AVAILABLE COLUMNS" list. The "AVAILABLE COLUMNS" list contains a large number of columns, including "smoker", "drink_level", "dress_preference", "ambience", "transport", "marital_status", "hijos", "birth_year", "interest", "personality", "religion", "activity", "color", "weight", "budget", and "height". Below this list, it says "19 columns available". To the right of the "AVAILABLE COLUMNS" list is a search bar with the placeholder "search columns" and a magnifying glass icon. Between the "AVAILABLE COLUMNS" and "SELECTED COLUMNS" lists is a large red arrow pointing to the right-pointing " > " button. The "SELECTED COLUMNS" list is currently empty, indicated by the message "0 columns selected". At the bottom right of the dialog box is a checkmark icon.

- Now click on Tick sign at the bottom of the page.

Select columns

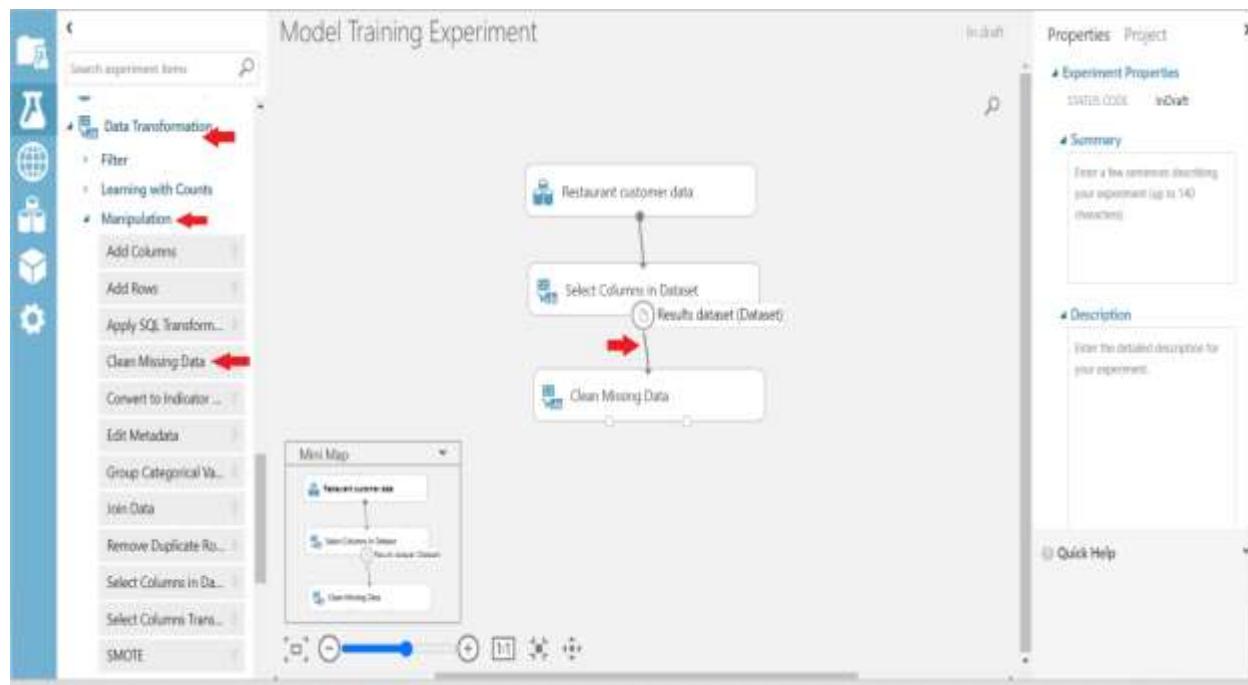
AVAILABLE COLUMNS

SELECTED COLUMNS

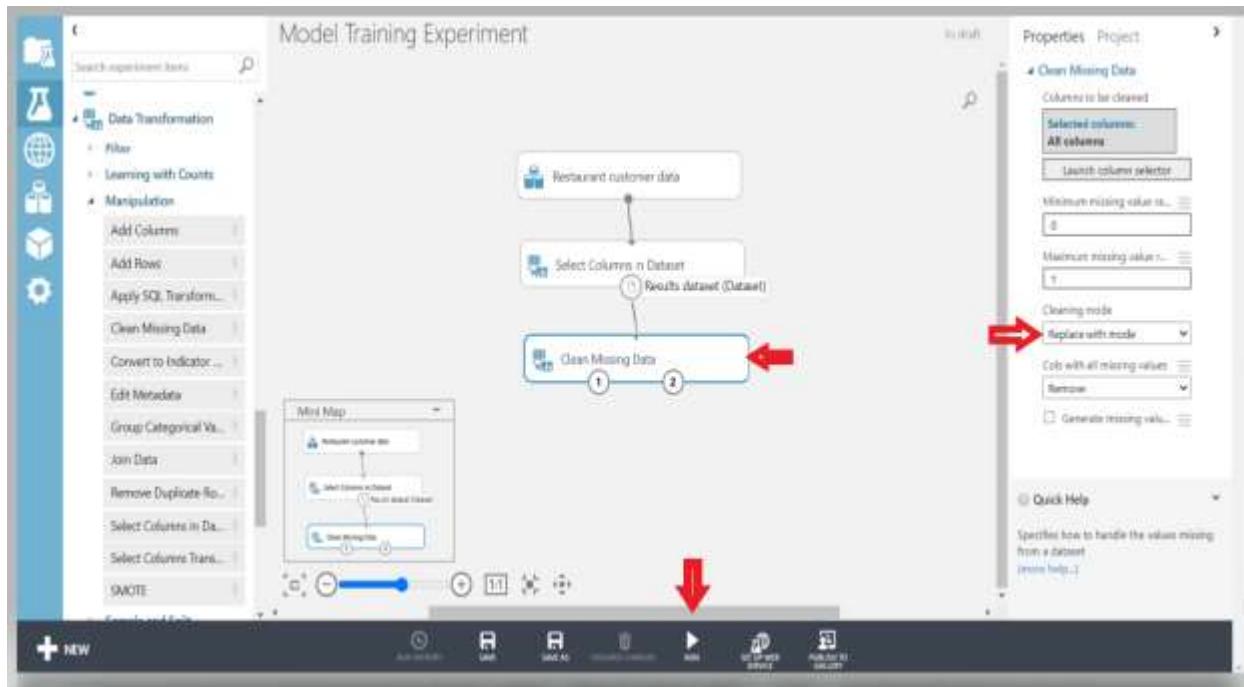
7 columns available | 12 columns selected

Selected

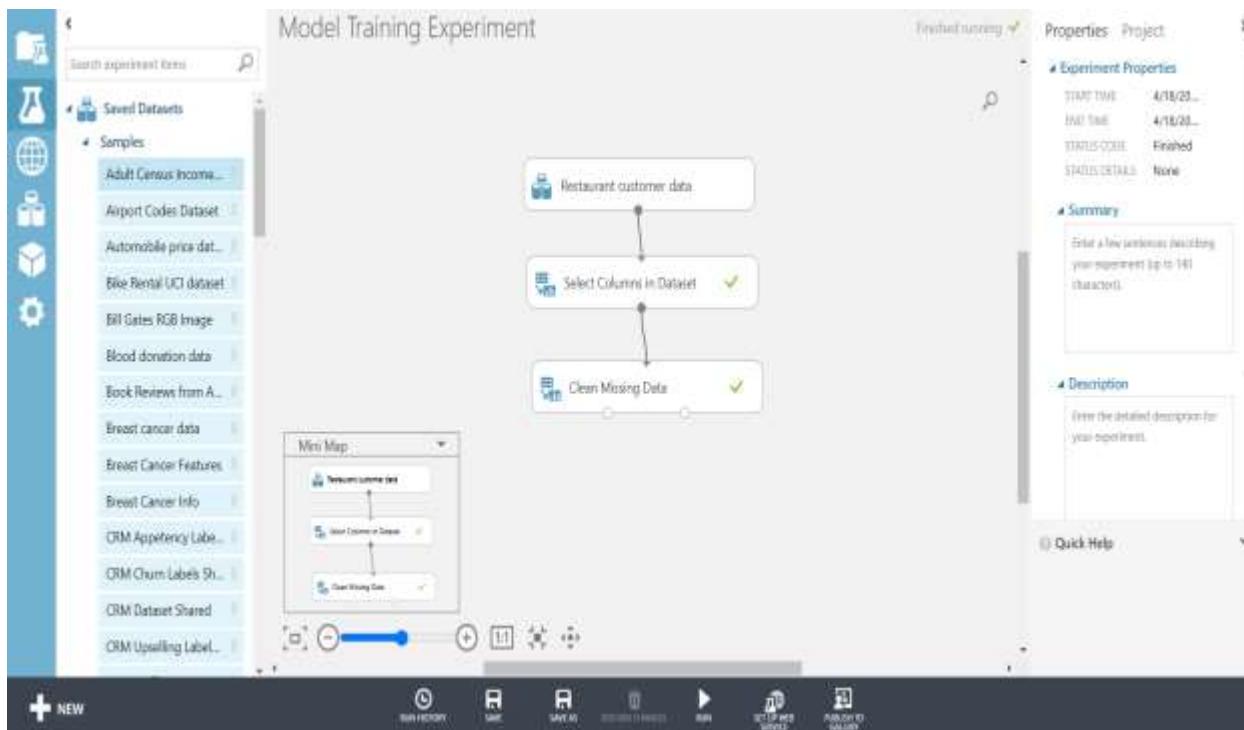
- Select the Data Transformation then switch to Manipulation then switch to Clean missing data, drag it into Panel and make connection.



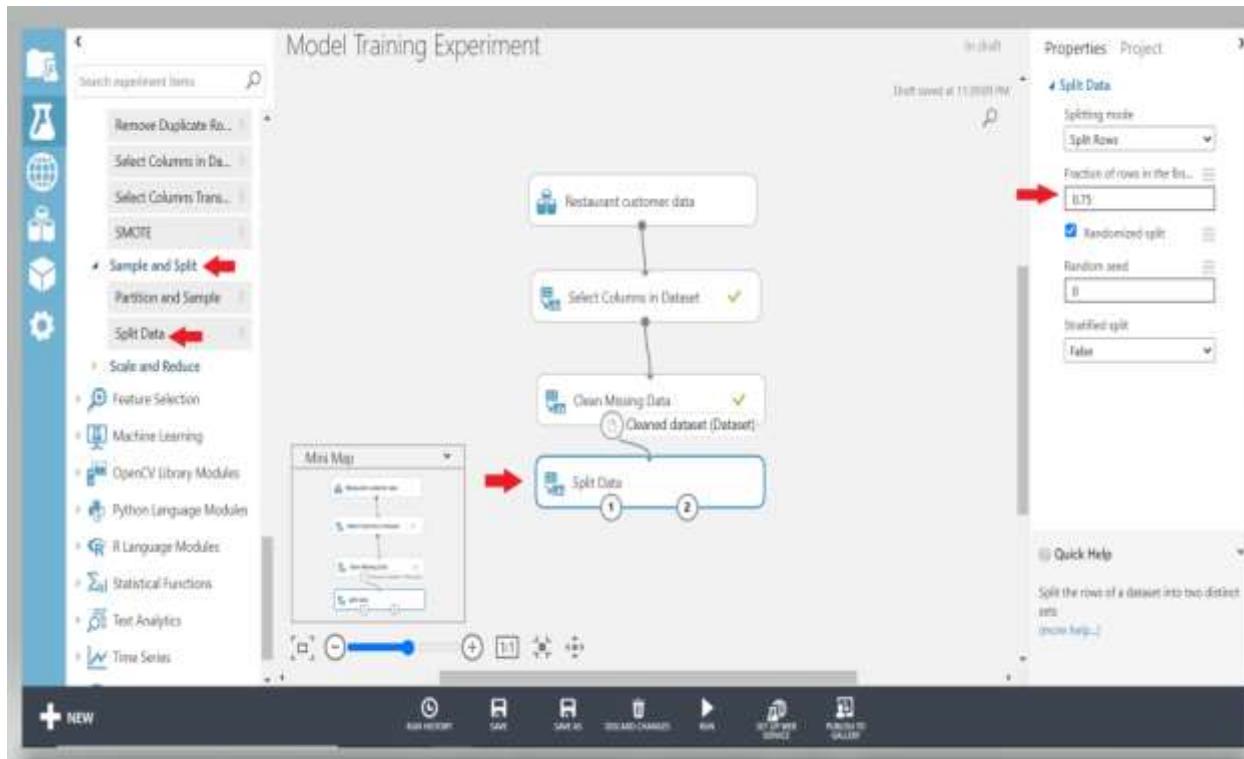
- Click on Clean missing data, set Replace with mode in cleaning mode and press Run.



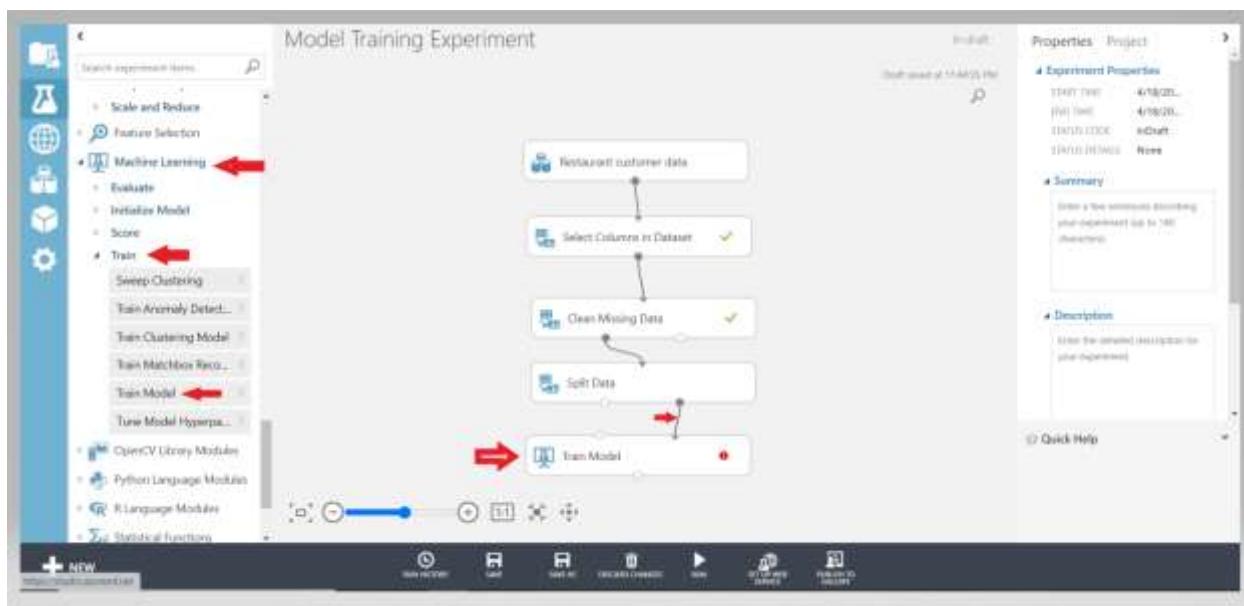
- If there is green tick, it means no error.



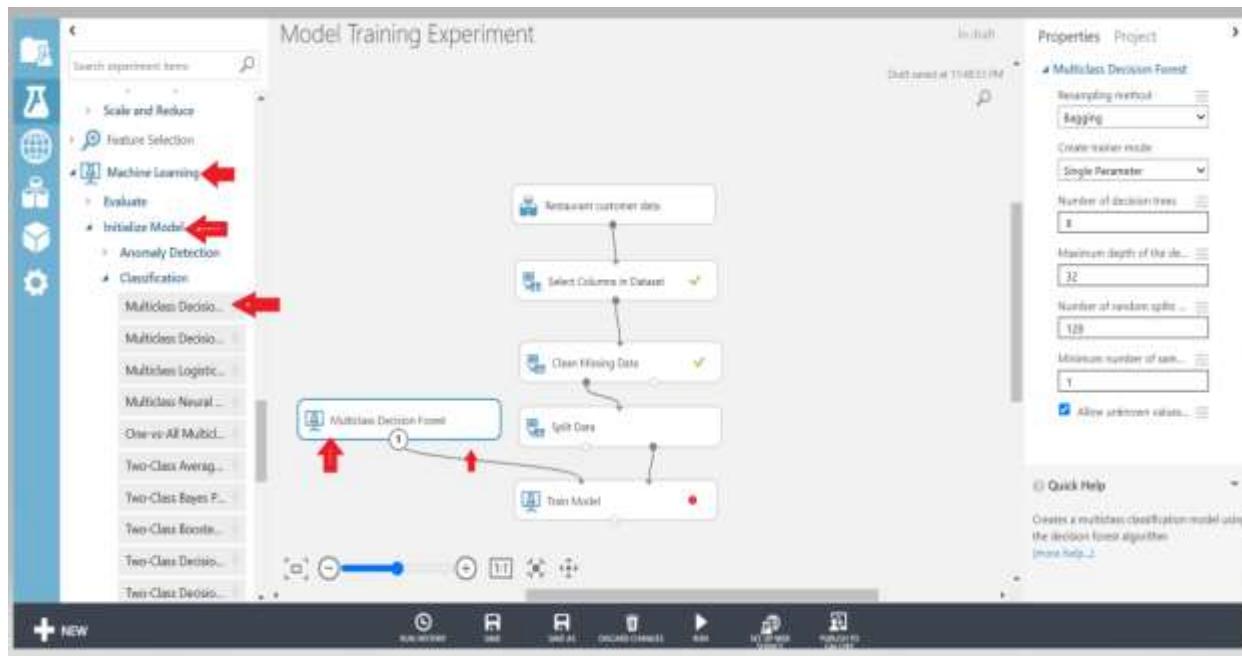
- Select Data Transformation then switch to Sample and Split then switch to Split data and drag it into Panel. Select the value 0.75 to split the data.



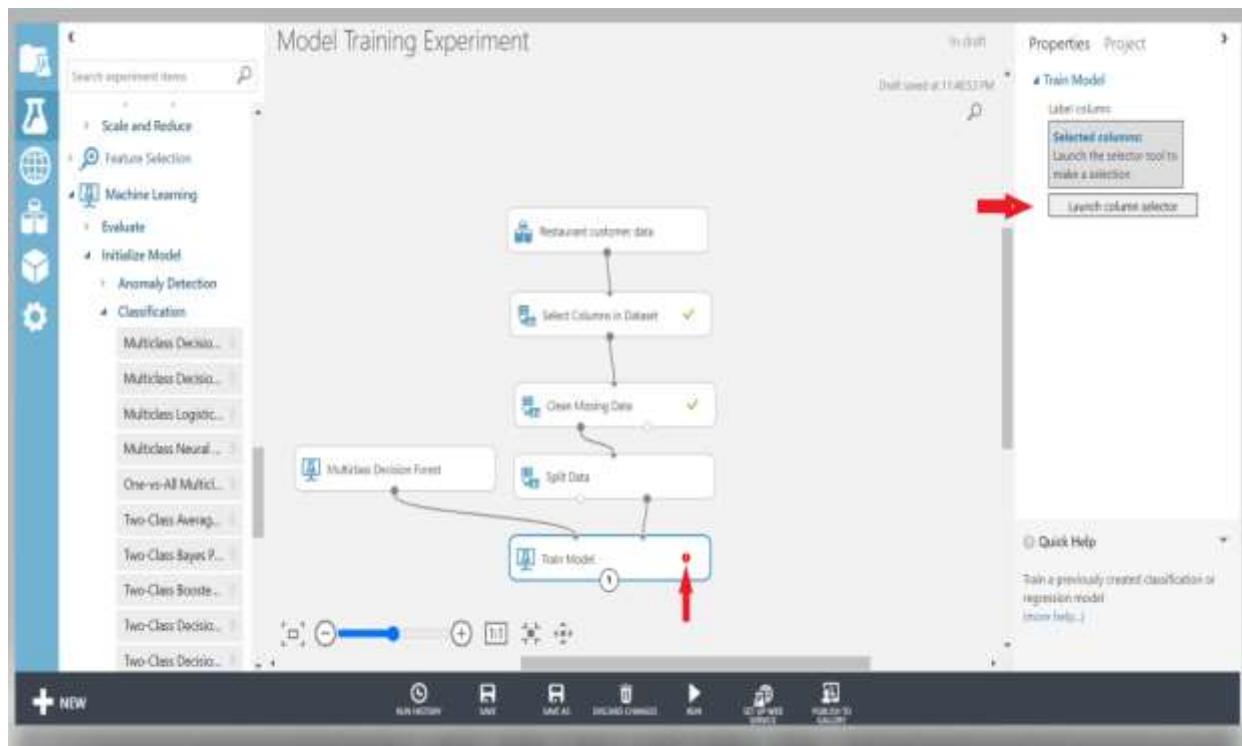
- Select Machine Learning



- Select the machine learning Algorithm from Machine Learning



- Click on Train Model and Launch Column Selector



- Now Select Budget as output for the prediction and press the tick mark.

Select a single column

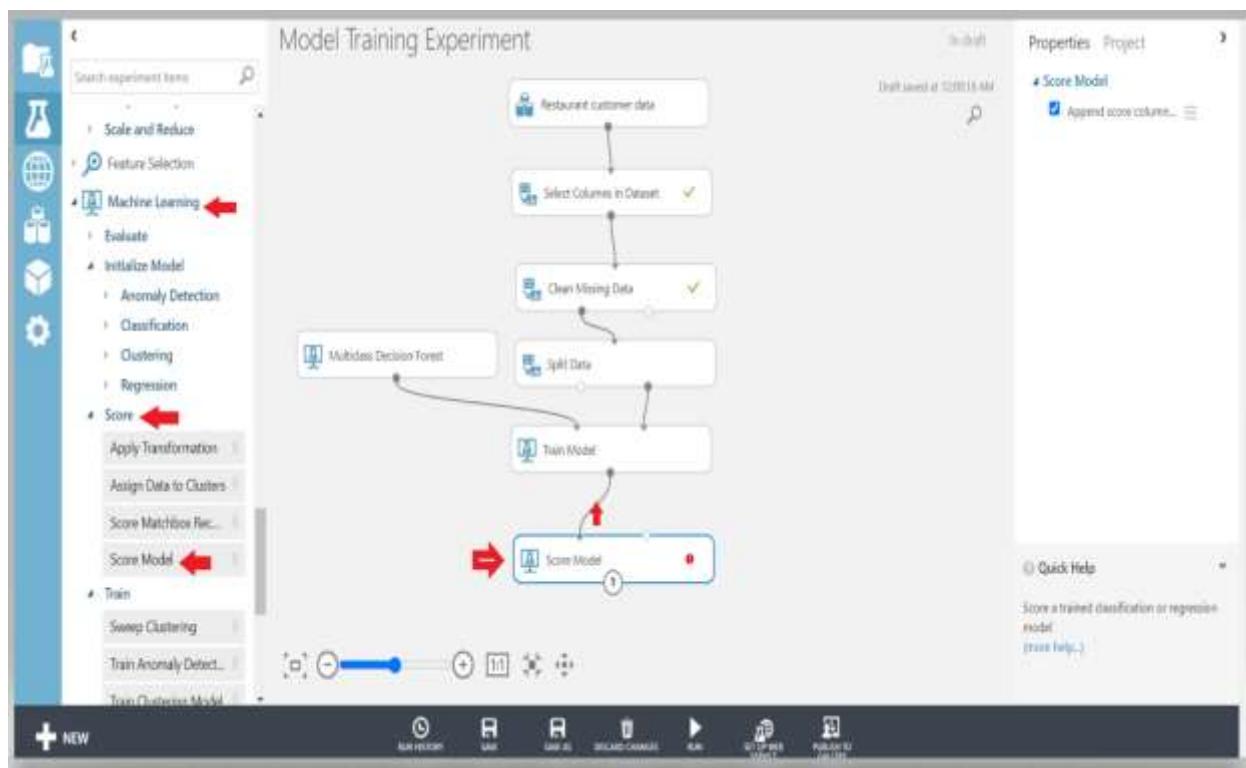
BY NAME

WITH RULES

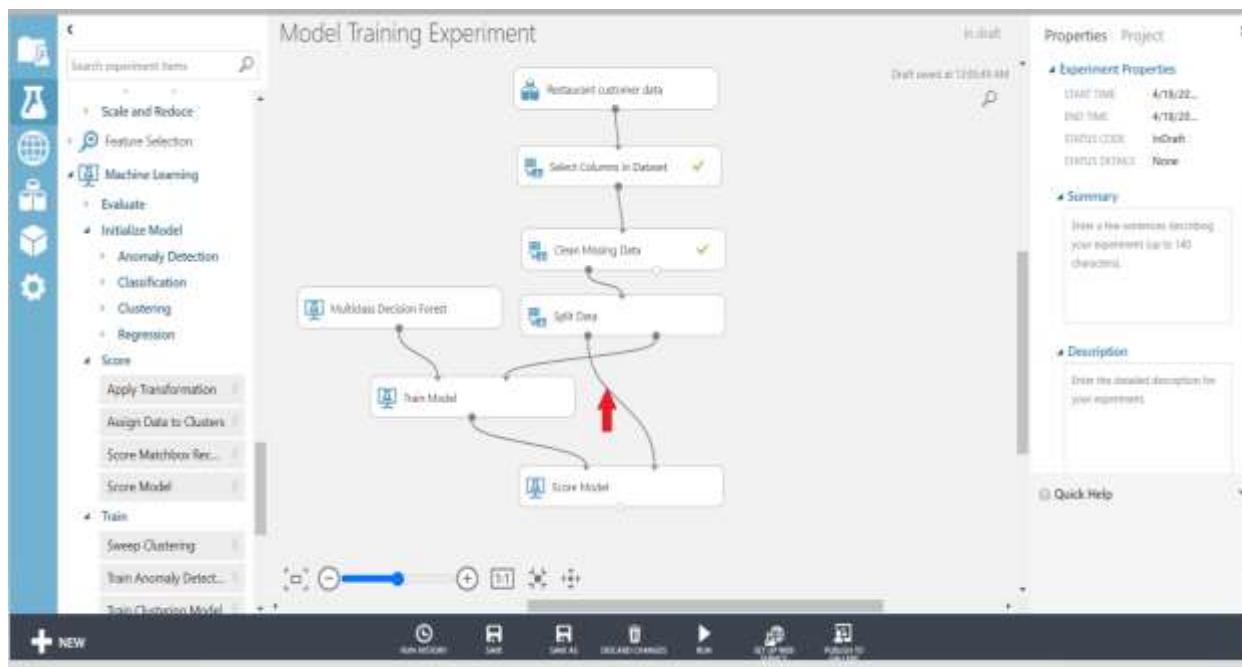
Include column names

smoker
drink_level
dress_preference
ambience
transport
mental_status
hijos
birth_year
interest
personality
activity
budget

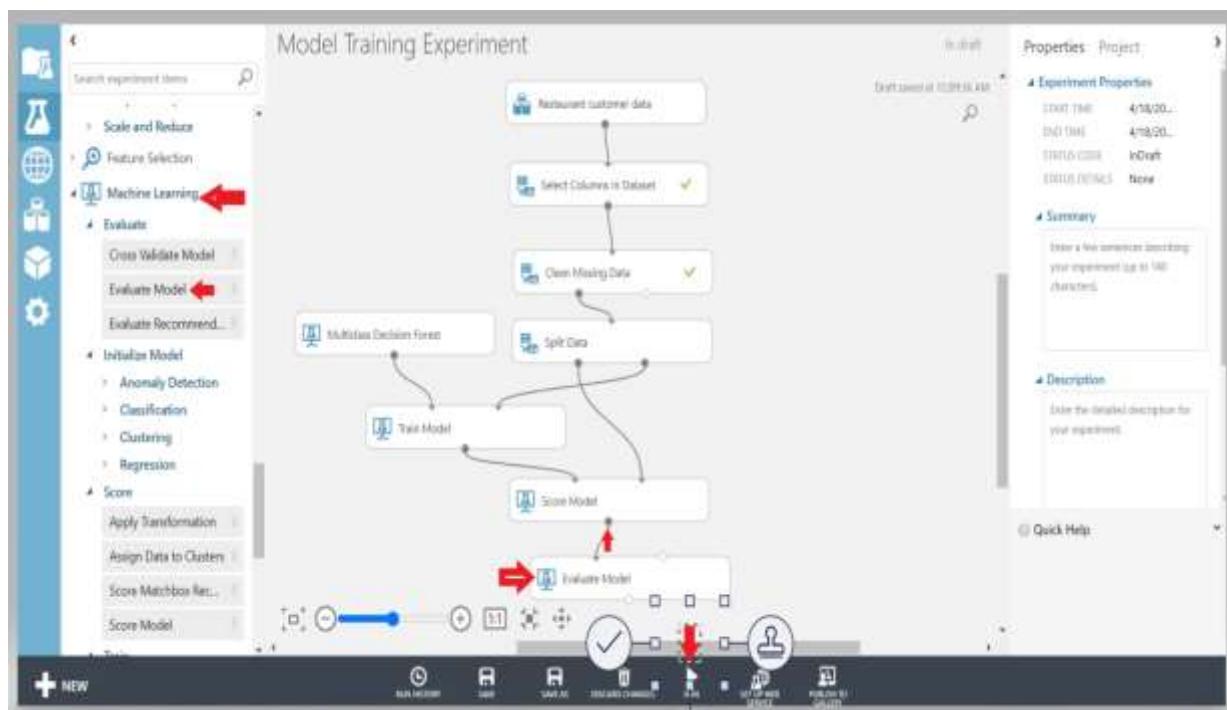
- For Score calculation, Choose Machine Learning then switch to Score then switch to Score Model and drag it into panel and connect it.



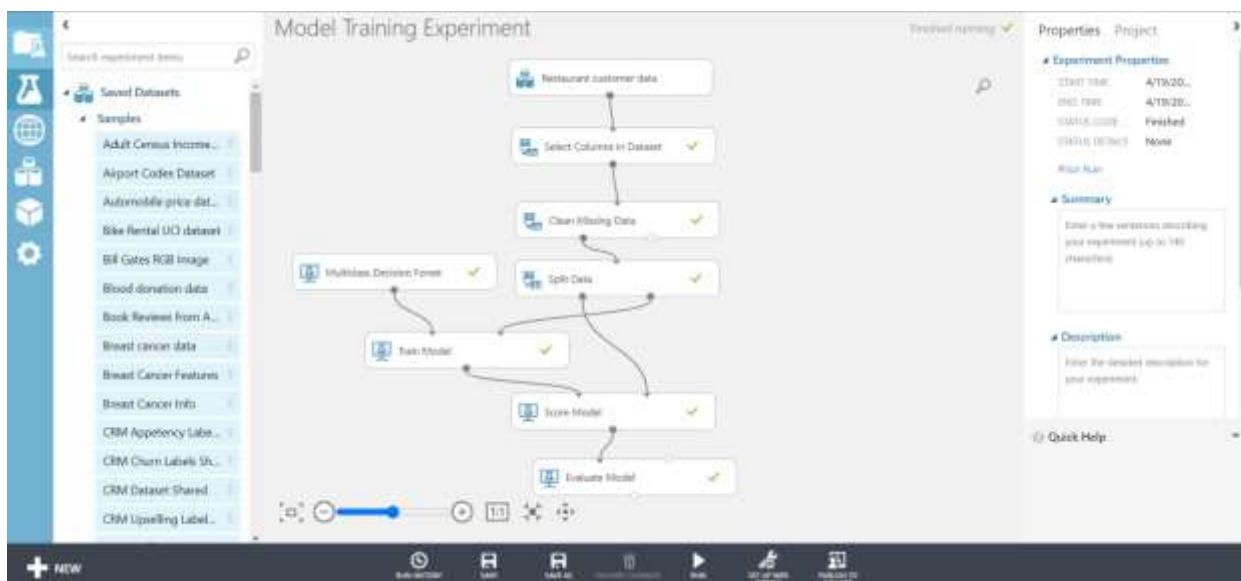
- Connect Split Data with Score Model for Testing.



- For Evaluation of the model choose, Machine Learning then switch to Evaluate then switch to Evaluate Model, drag it into panel and connect.



- Now Run the model. Green tick shows that model has run successfully.



- To see the visualization Result, right click on Evaluate model then Visualize

Model Training Experiment > Evaluate Model > Evaluation results:

Average accuracy	0.775641
Micro-averaged precision	0.663462
Macro-averaged precision	0.341758
Micro-averaged recall	0.663462
Macro-averaged recall	0.335759

Confusion Matrix



Bare Coding

Machine Learning is one of the fastest-growing fields which has witnessed exponential growth in the technical world. There is no best language for machine learning it depends on what you want to build, to work in this field, you just need to learn only one particular programming language very well based on your own comfort, project requirements, and predilections. Just explore some of these mostly used

languages and pick up one of your choices, you don't need to take anyone's recommendation.

1. Python

Python leads all the other languages with more than 60% of machine learning developers using and prioritizing it for development because python is easy to learn. Scalable and open source. Python has many awesome visualization packages and useful core libraries like NumPy, SciPy, pandas, matplotlib, seaborn, sklearn which really make your work very easy and empower the machines to learn.

- **NumPy:** Numeric Python or NumPy is a Linear Algebra Library for Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices.
- **Pandas:** It is the most popular Python library which provides highly optimized performance for data analysis.
- **Matplotlib:** It is a popular python plotting library used for creating basic graphs like line charts, bar charts, histograms, and many more.
- **Seaborn:** Provides a high-level interface for creating attractive graphs
- **sci-kit Learn:** It is used for data mining and data analysis which implements a wide-range of machine-learning algorithms like classification, regression, and clustering algorithms including support vector machines, random forests, gradient boosting, k-means.

2. Java

This programming language is the “Jack of all the trade” and continues to dominate over in the ML industry also. Java provides many good environments like Weka, Knime, RapidMiner, Elka which used to perform machine learning tasks using graphical user interfaces.

- **Weka:** It is a free, portable library primarily used for data mining, data analysis, and predictive modelling and best used for machine learning algorithms. it is easy to use with the graphical interface and supports several standard data mining tasks, including data preprocessing, classification, clustering, and feature selection.
- **JavaML:** A Java API with simple and easy interfaces to implement the collection of machine learning and data mining algorithms in Java with clearly written and properly documented implementation of algorithms.
- **Deeplearning4j:** It is an innovative open-source distributed deep learning library that provides a computing framework with wide support for machine learning algorithms. This library is extremely useful for identifying patterns,

sentiment, sound, and text and is designed especially for business environments.

- **ELKI:** It is a unique open-source data mining framework that mainly focused in the independent evaluation of data mining algorithms and data management and emphasizes in unsupervised methods. It also allows arbitrary data types, file formats, or distance or similarity measures.

3. C++

The superfast C++ programming language is also very popular in the field of machine learning. This powerful language gets supported by most of the machine learning platforms. If you have some good working knowledge using C++ then it is a pretty good idea to learn machine learning using C++. C++ is much efficient compare to most of programming languages. Many powerful libraries such as TensorFlow and Torch are implemented in the C++ programming language so machine learning and C++ is truly a great combination.

- **TensorFlow:** Google's open-source TensorFlow is used to do numerical computations on any CPU or GPU using data flow graphs and make decisions with whatever information it gets.
- **Torch:** A open-source machine learning library which makes scientific and numerical operation easier by providing a large number of algorithms. it makes for easier and improved efficiency and speed.
- **mlpack:** A superfast, flexible machine learning library which provides fast and extensible implementations of cutting-edge machine learning algorithms using C++ classes which can be integrated into larger-scale machine learning solutions

4. R

R is a very popular programming language for statistical computing, analysis, and visualizations in machine learning. It is a perfect graphics-based language for exploring the statistical data via graph vastly used by data professionals at Facebook, Google, etc. Though R is highly preferable in bioengineering and biomedical statistic it is also popular in implementing machine learning like regression, classification, and decision tree formation.

- **xgboost:** this is used for implementing the gradient boosting framework and is popular for it's performance and speed. It supports various objective functions like regression, classification, and ranking and is extensible so that you can define your own objectives easily.
- **mlr:** It is an extensible framework for classification, regression, and clustering problems and has easy extension mechanism through s3 inheritance.

- **PARTY:** this package is used for recursive partitioning. This package is used to build decision trees based on the Conditional Inference algorithm. This package is also extensive, which reduces the training time and bias.
- **CARET:** this package is developed to combine model training and prediction for several different algorithms for a given business problem and helps to choose the best machine learning algorithm.

5. Javascript

It is one of the most widely used, high-level, and dynamically typed languages which is flexible and multi-paradigm. Javascript is also so popular in ML that high-profile projects like Google's Tensorflow.js are based on JavaScript. If you are a master of Javascript then literally you can do everything from full-stack to machine learning and NLP.

- **Brain.js:** It is a GPU accelerated, easy to integrate neural networks in JavaScript which is used with Node.js in the browser and provides multiple neural network implementations to train to do different things well. It is so simple to use that you do not need to know Neural Networks in detail to work with this.
- **Tensorflow.js:** It is a popular library for machine learning in JavaScript. You can build and train models directly in JavaScript using flexible APIs and almost any problems in Machine Learning can be solved using Tensorflow.js. You can also retrain the existing ML models using your own data.
- **machinelearn.js:** It is the savior of Javascript which is a replacement of python's ScikitLearn library. It provides clustering, decomposition, feature extractions models, and utilities for supervised and unsupervised learning.
- **face-api.js:** A ready-to-use APIs that includes implementations of well-known Face Detection and Recognition models which is pre-trained with a wide variety of datasets. It gives you the flexibility to directly plug into any Node.js and browser environments. Being lightweight this library can be used on both mobile and web browsers with no issues.

Among these programming languages, **Python** remains the most popular in the field of ML.

1.10 Coding Platform: Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore

reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

Released in 1991, Python is used for;

- Web development (server-side),
- Software development,
- Mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax like the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Python Syntax compared to other programming languages

- Python was designed for readability and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

1.11 Anaconda: Introduction & Installation

Anaconda is a free open-source data science tool that focusses on the distribution of R and Python programming languages for data science and machine learning tasks. Anaconda aims at simplifying the data management and deployment of the same.

Anaconda is popular because it brings many of the tools used in data science and machine learning with just one install, so it's great for having short and simple setup. Anaconda also uses the concept of creating environments so as to isolate different libraries and versions.

Anaconda is a powerful data science platform for data scientists. The package manager of Anaconda is the conda which manages the package versions. Anaconda is a tool that offers all the required package involved in data science at once. The programmers choose Anaconda for its ease of use.

Anaconda is written in Python, and the worthy information on Conda is unlike pip in Python, this package manager checks for the requirement of the dependencies and installs it if it is required. More importantly, warning signs are given if the dependencies already exist.

Conda very quickly installs the dependencies along with frequent updates. It facilitates creation and loading with equal speed along with easy environment switching. Anaconda is pre-built with more than 1500 Python or R data science packages. Anaconda has specific tools to collect data using Machine learning and Artificial Intelligence. Anaconda is indeed a tool used for developing, testing and training in one single system. The tool can be managed with any project as the environment is easily manageable. The installation of Anaconda is very easy and most preferred by non-programmers who are data scientists.

Comparison Table Between Anaconda and Python

Parameter of Comparison	Anaconda	Python
Definition	Anaconda is the enterprise data science platform which distributes R and Python for machine learning and data science	Python is a high-level general-purpose programming language used for machine learning and data science
Category	Anaconda belongs to Data Science Tools	Python belongs to Computer Languages

Package Manager	Anaconda has conda has its package manager	Python has pip as the package manager
User Applications	Anaconda is primarily developed to support data science and machine learning tasks	Python is not only used in data science and machine learning but also a variety of applications in embedded systems, web development, and networking program
Package Management	Package manager conda allows Python as well as Non-Python library dependencies to install.	Package manager pip allows all the Python dependencies to install

Anaconda is a new distribution of the Python and R data science package. It was formerly known as Continuum Analytics. Anaconda has more than 100 new packages. This work environment, Anaconda is used for scientific computing, data science, statistical analysis, and machine learning.

Anaconda Individual Edition contains conda and Anaconda Navigator, as well as Python and hundreds of scientific packages. When you installed Anaconda, you installed all these too.

Conda works on your command line interface such as Anaconda Prompt on Windows and terminal on macOS and Linux.

Installing Anaconda on Windows OS

Anaconda is an open-source software that contains Jupyter, spyder, etc that are used for large data processing, data analytics, heavy scientific computing. Anaconda works for R and python programming language.

1. At first, visit the following link: <https://www.anaconda.com/distribution/> and the page will pop up like this.

The screenshot shows the Anaconda Distribution website. At the top, there's a navigation bar with links for Products, Pricing, Solutions, Resources, Partners, Blog, Company, and Contact Sales. Below the navigation, a green banner reads "Individual Edition is now ANACONDA DISTRIBUTION". A sub-headline says "The world's most popular open-source Python distribution platform". To the right, there's a call-to-action button labeled "Anaconda Distribution" with a "Download" link and a Windows icon. Below it, a section for "For Windows" offers a "Python 3.9 * 64-Bit Graphical Installer" (594 MB). Further down, there's a "Get Additional Installers" section with icons for Windows, macOS, and Linux.

2. Scroll down the page and select windows.



3. Download Python 3.7 or Above Version (Recommended) as Python version 2 will have no more support by the community at the end of 2019. Depending on your computer system, choose either 32-bit or 64-bit installer to download the .exe file.

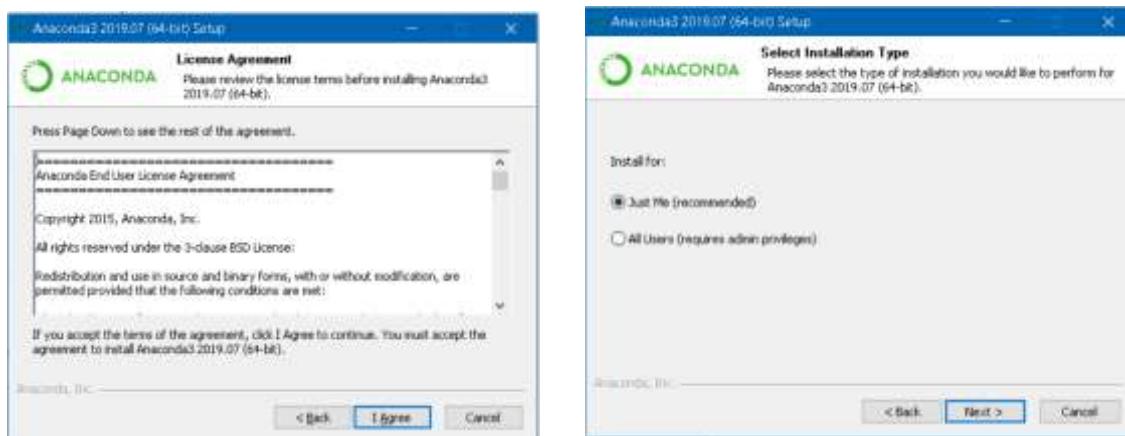
The screenshot shows the "Anaconda Installers" page. It has three main sections: "Windows", "macOS", and "Linux". The "Windows" section shows two download links: "Python 3.9 64-Bit Graphical Installer (594 MB)" and "32-Bit Graphical Installer (488 MB)", with the first one highlighted by a red box. The "macOS" section lists four download links: "Python 3.9 64-Bit Graphical Installer (591 MB)", "64-Bit Command Line Installer (584 MB)", "64-Bit (M1) Graphical Installer (428 MB)", and "64-Bit (M1) Command Line Installer (420 MB)". The "Linux" section lists three download links: "Python 3.9 64-Bit (x86) Installer (659 MB)", "64-Bit (Power8 and Power9) Installer (367 MB)", and "64-bit (AWS Graviton2 / ARM64) Installer (568 MB)".

4. After downloading the file, run the file. The file will open, Click **Next**

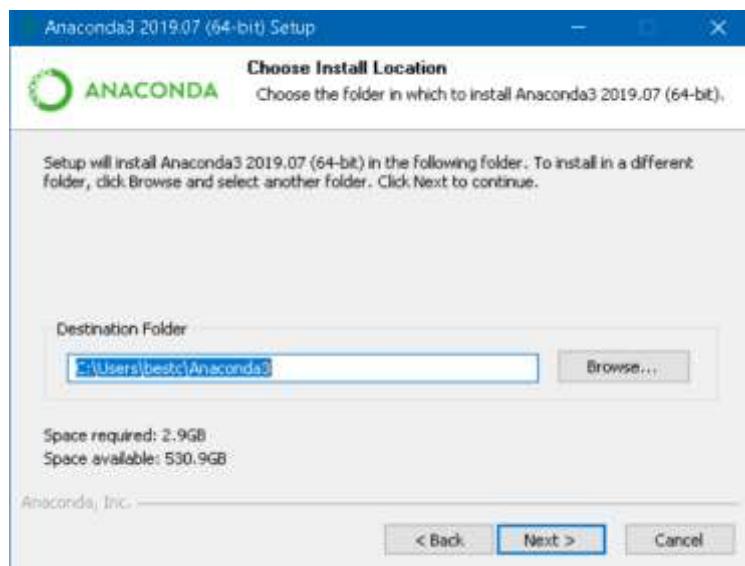


5. And click **I Agree** to the license.

6. Choose **Just Me** and click **Next**



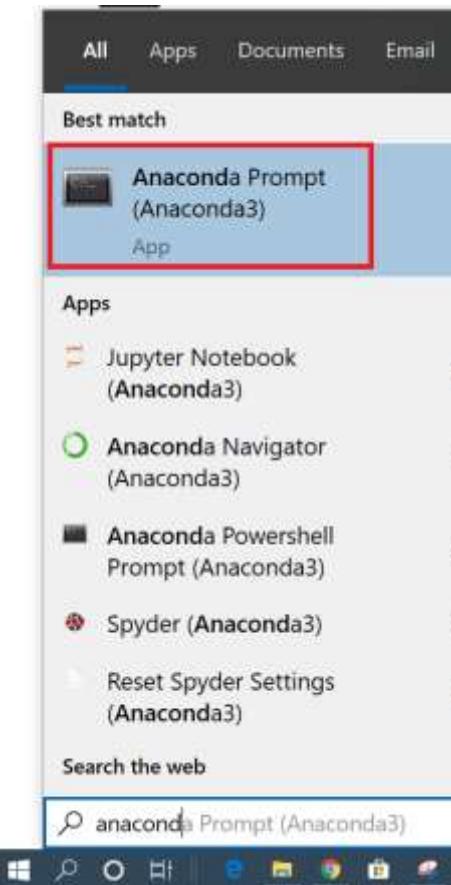
7. Choose the installation location by clicking **Browse** or leave it as it is (default location) and continue to click **Next**.



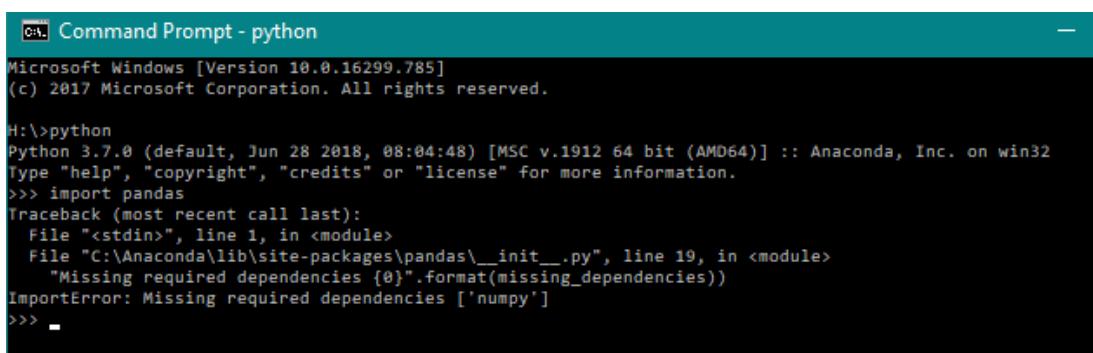
8. Here, it is highly recommended to choose the second one “**Register Anaconda as my default Python 3.7**” and click **Install**



9. Once the installation is done, open the **Anaconda Prompt** from Windows start menu bar.



10. Anaconda Prompt is shell like Windows Command Prompt (Windows Terminal) powered by Anaconda distribution. To check whether we have successfully installed Anaconda or not, type **python** command in the shell.



```
PS C:\> python
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
(c) 2017 Microsoft Corporation. All rights reserved.

H:\>python
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "C:\Anaconda\lib\site-packages\pandas\_init_.py", line 19, in <module>
      "Missing required dependencies {0}".format(missing_dependencies))
ImportError: Missing required dependencies ['numpy']
>>>
```

Installing Anaconda on Linux Kernel

1. At first, visit the following link: <https://www.anaconda.com/distribution/> and the page will pop up like this.
2. Scroll down the page and select windows.



3. Once it is downloaded go to Download folder and run .sh file

```
Karthick@LinuxShellTips:~/Downloads$ ls -l
total 557480
-rw-rw-r-- 1 Karthick Karthick 570853747 Jun 22 18:39 Anaconda3-2021.05-Linux-x86_64.sh
Karthick@LinuxShellTips:~/Downloads$ sha256sum Anaconda3-2021.05-Linux-x86_64.sh
2751ab3d678ff0277ae80f9e8a74f218fcf70fe9a9cdc7bb1c137d7e47e33d53 Anaconda3-2021.05-Linux-x86_64.sh
```

4. Now run the downloaded .sh file to install anaconda. As a first step, it will ask you to read the license agreement once you press enter

```
$ bash Anaconda3-2021.05-Linux-x86_64.sh
```

```
Karthick@LinuxShellTips:~/Downloads$ bash Anaconda3-2021.05-Linux-x86_64.sh
Welcome to Anaconda3 2021.05

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

- In the next step, it will ask you to choose a location where the anaconda will be installed. It defaults to your home directory.

```
Anaconda3 will now be installed into this location:  
/home/karthick/anaconda3
```

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

```
[/home/karthick/anaconda3] >>> █
```

- Packages will be installed and once the installation is completed it will ask to initialize **Anaconda3** by running **conda init**. It defaults to **No**. You can choose **Yes** or **No** depending upon how you need it.

```
Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>>

You have chosen to not have conda modify your shell scripts at all.
To activate conda's base environment in your current shell session:

eval "$(/home/karthick/anaconda3/bin/conda shell.YOUR_SHELL_NAME hook)"

To install conda's shell functions for easier access, first activate, then:

conda init

If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

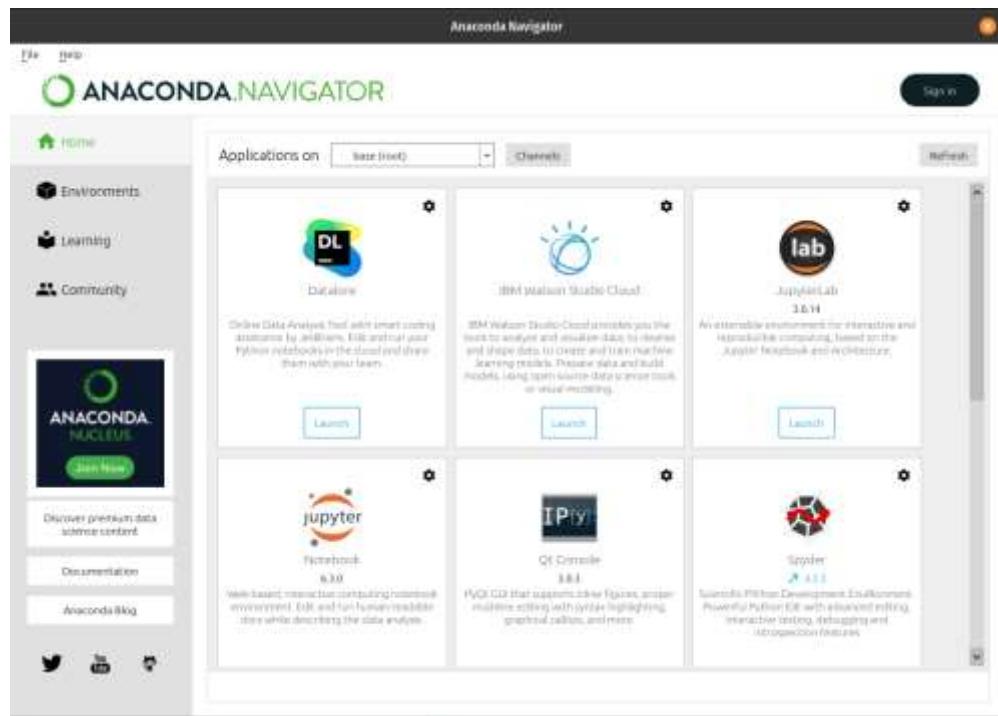
Thank you for installing Anaconda3!

=====
Working with Python and Jupyter notebooks is a breeze with PyCharm Pro,
designed to be used with Anaconda. Download now and have the best data
tools at your fingertips.

PyCharm Pro for Anaconda is available at: https://www.anaconda.com/pycharm
```

- Go to the directory where **anaconda** is installed and under the **bin** directory, there is a binary called “**anaconda-navigator**”. This will launch the GUI program for anaconda from where you can launch your tools.

```
$ ./home/karthick/anaconda3/bin/anaconda-navigator
```



Jupyter notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at [Project Jupyter](#).

IF you already installed Anaconda in your machine then its very easy to use Jupyter notebook

Just Run command from Anaconda to open Jupyter notebook

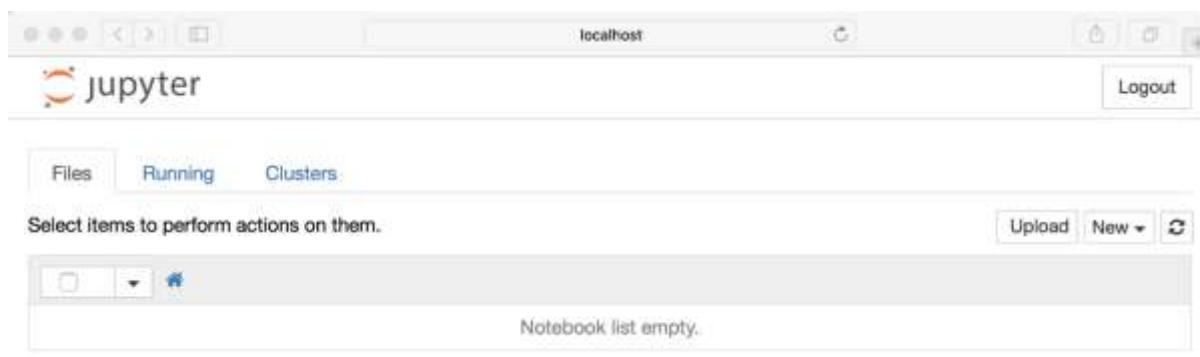
```
Anaconda Prompt (Anaconda3) - Jupyter notebook:
(base) C:\Users\PRAVIN>jupyter notebook
[1 14:38:04.695 NotebookApp] JupyterLab extension loaded from C:\ProgramData\Anaconda3\lib\site-packages\jupyterlab
[1 14:38:04.695 NotebookApp] JupyterLab application directory is C:\ProgramData\Anaconda3\share\jupyter\lab
[1 14:38:04.698 NotebookApp] Serving notebooks from local directory: C:\Users\PRAVIN
[1 14:38:04.698 NotebookApp] Jupyter Notebook 6.1.4 is running at:
[1 14:38:04.698 NotebookApp] http://localhost:8888/?token=fce81d78fb022669006757133ffae92129775d35581a8513
[1 14:38:04.699 NotebookApp] or http://127.0.0.1:8888/?token=fce81d78fb022669006757133ffae92129775d35581a8513
[1 14:38:04.699 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:38:04.796 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/PRAVIN/AppData/Roaming/jupyter/runtime/nbserver-11204-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=fce81d78fb022669006757133ffae92129775d35581a8513
or http://127.0.0.1:8888/?token=fce81d78fb022669006757133ffae92129775d35581a8513
```

Jupyter notebook will open in your default browser, should start (or open a new tab)

to the following URL: <http://localhost:8888/tree>

Your browser should now look something like this:

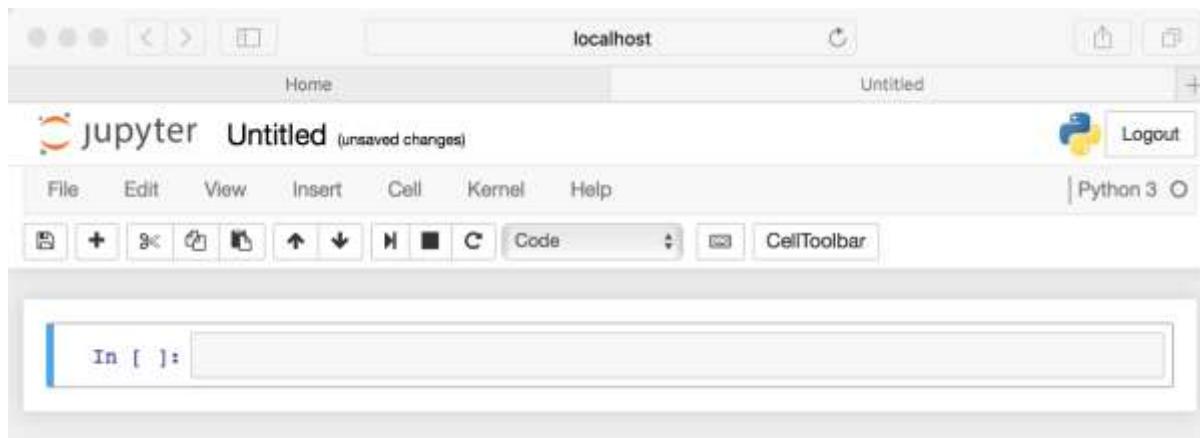


Creating a Notebook

Now that you know how to start a Notebook server, you should probably learn how to create an actual Notebook document.

All you need to do is click on the *New* button (upper right), and it will open up a list of choices. On my machine, I happen to have Python 2 and Python 3 installed, so I can create a Notebook that uses either of these. For simplicity's sake, let's choose Python 3.

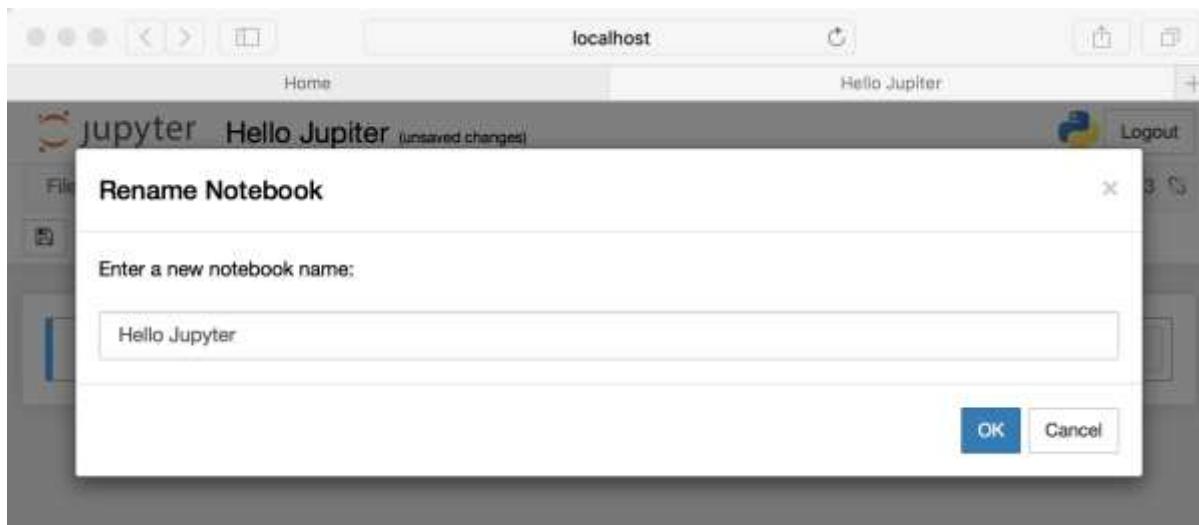
Your web page should now look like this:



Naming

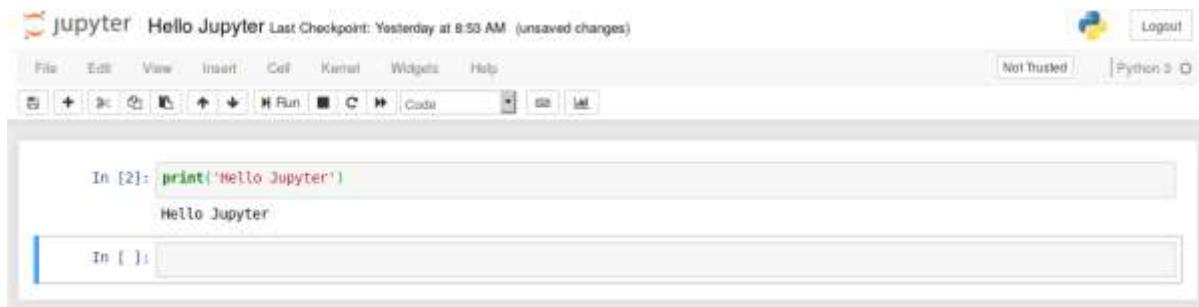
You will notice that at the top of the page is the word *Untitled*. This is the title for the page and the name of your Notebook. Since that isn't a very descriptive name, let's change it!

Just move your mouse over the word *Untitled* and click on the text. You should now see an in-browser dialog titled *Rename Notebook*. Let's rename this one to *Hello Jupyter*.



Running Cells

Running a cell means that you will execute the cell's contents. To execute a cell, you can just select the cell and click the *Run* button that is in the row of buttons along the top. It's towards the middle. If you prefer using your keyboard, you can just press **Shift + Enter**.



If you have multiple cells in your Notebook, and you run the cells in order, you can share your variables and imports across cells. This makes it easy to separate out your code into logical chunks without needing to reimport libraries or recreate variables or functions in every cell.

The Menus

The Jupyter Notebook has several menus that you can use to interact with your Notebook. The menu runs along the top of the Notebook just like menus do in other applications. Here is a list of the current menus:

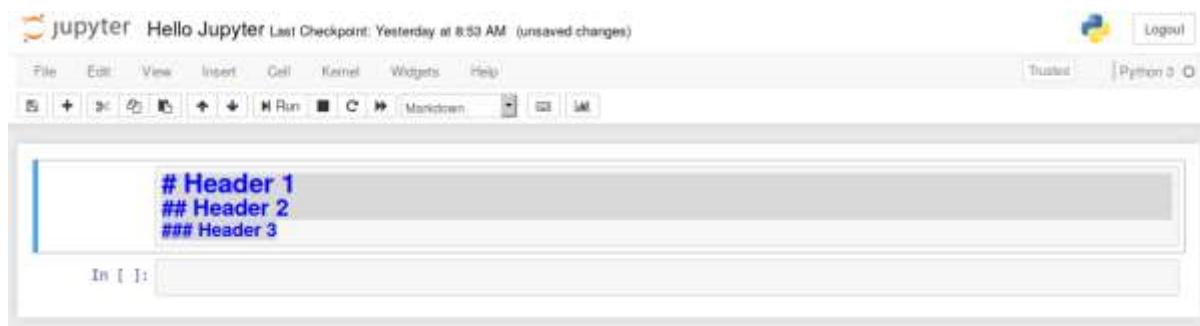
- File
- Edit
- View
- Insert
- Cell
- Kernel

- Widgets
- Help

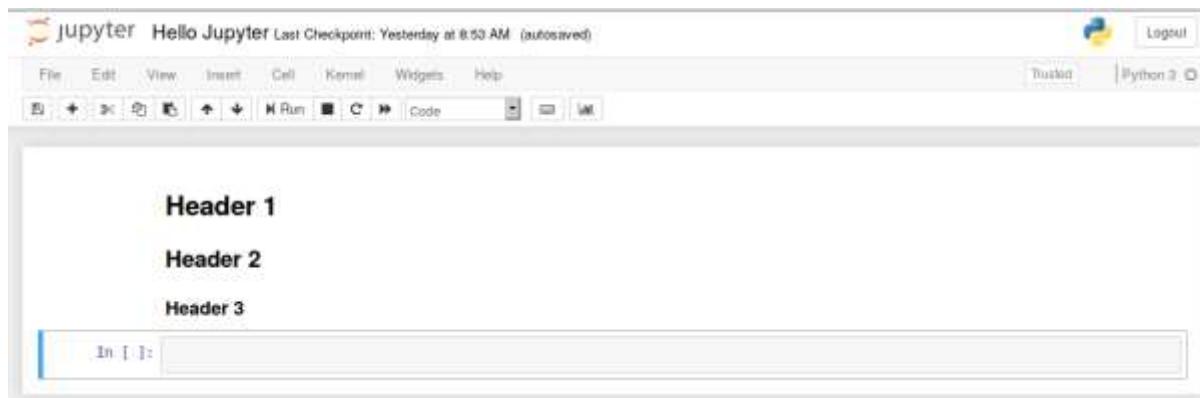
Explore all the Menu one by one to Use of jupyter notebook.

Headers

Creating headers in Markdown is also quite simple. You just have to use the humble pound sign. The more pound signs you use, the smaller the header. Jupyter Notebook even kind of previews it for you:



Then when you run the cell, you will end up with a nicely formatted header:



Exporting Notebooks

When you are working with Jupyter Notebooks, you will find that you need to share your results with non-technical people. When that happens, you can use the nbconvert tool which comes with Jupyter Notebook to convert or export your Notebook into one of the following formats:

- HTML
- LaTeX
- PDF
- RevealJS
- Markdown
- ReStructured Text

- Executable script

The nbconvert tool uses Jinja templates under the covers to convert your Notebook files (.ipynb) into these other formats.

1.12 Requisite libraries: Numpy, Pandas & Seaborn

1.12.1 NumPy

NumPy is a module for Python that allows you to work with multidimensional arrays and matrices. It's perfect for scientific or mathematical calculations because it's fast and efficient. In addition, NumPy includes support for signal processing and linear algebra operations. So if you need to do any mathematical operations on your data, NumPy is probably the library for you.

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

NumPy in Python is a library that is used to work with arrays and was created in 2005 by Travis Oliphant. NumPy library in Python has functions for working in domain of Fourier transform, linear algebra, and matrices. Python NumPy is an open-source project that can be used freely. NumPy stands for Numerical Python.

How to install NumPy Python?

Installing the NumPy library is a straightforward process. You can use pip to install the library. Go to the command line and type the following:

pip install numpy

If you are using Anaconda distribution, then you can use conda to install NumPy. conda install numpy Once the installation is complete, you can verify it by importing the NumPy library in the python interpreter. One can use the numpy library by importing it as shown below.

If the import is successful, then you will see the following output.

```
>>> import numpy  
>>> numpy.__version__
```

1.12.2 Pandas Library

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

Advantages

- Fast and efficient for manipulating and analyzing data.
- Data from different file objects can be loaded.
- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data
- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
- Data set merging and joining.
- Flexible reshaping and pivoting of data sets
- Provides time-series functionality.
- Powerful group by functionality for performing split-apply-combine operations on data sets.

Getting Started

The first step of working in pandas is to ensure whether it is installed in the Python folder or not. If not then we need to install it in our system using **pip command**. Type cmd command in the search box and locate the folder using cd command where **python-pip file** has been installed. After locating it, type the command:

pip install pandas

After the pandas have been installed into the system, you need to import the library. This module is generally imported as:

import pandas as pd

Here, pd is referred to as an alias to the Pandas. However, it is not necessary to import the library using the alias, it just helps in writing less amount code every time a method or property is called.

Pandas generally provide two data structures for manipulating data, They are:

- **Series**
- **DataFrame**

1.12.3 Seaborn Library

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

Plots are basically used for visualizing the relationship between variables. Those variables can be either be completely numerical or a category like a group, class or division. Seaborn divides plot into the below categories –

- **Relational plots:** This plot is used to understand the relation between two variables.
- **Categorical plots:** This plot deals with categorical variables and how they can be visualized.
- **Distribution plots:** This plot is used for examining univariate and bivariate distributions
- **Regression plots:** The regression plots in seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.
- **Matrix plots:** A matrix plot is an array of scatterplots.
- **Multi-plot grids:** It is an useful approach is to draw multiple instances of the same plot on different subsets of the dataset.

Installation

For python environment :

pip install seaborn

For conda environment :

conda install seaborn

1.13 Scikit-learn: Bring ML frameworks in Python

Scikit-Learn is a free machine learning library for Python. It supports both supervised and unsupervised machine learning, providing diverse algorithms for classification, regression, clustering, and dimensionality reduction. The library is built using many libraries you may already be familiar with, such as NumPy and SciPy. It also plays well with other libraries, such as Pandas and Seaborn.

Installing Scikit-Learn can be done using either the pip package manager or the conda package manager. Simply write the code below into your command line editor or terminal and let the package manager handle the installation:

pip install sklearn

conda install sklearn

The package manager will handle installing any required dependencies for the Scikit-learn library you may not already have installed.

Try writing the script below and running it. If it runs without issue, then you successfully installed Scikit-learn

import sklearn

Key Features

Scikit-learn offers a variety of tools and algorithms for machine learning. Some of its key features include:

- Simple and consistent API: Scikit-learn provides a consistent interface for all its functions, making it easy to learn and use.
- Comprehensive documentation: The library is well-documented, with clear explanations and examples for each algorithm and function.
- Preprocessing tools: Scikit-learn offers tools for data preprocessing, including feature scaling, encoding categorical variables, and dimensionality reduction.
- Model selection and evaluation: The library provides methods for model selection and evaluation, such as cross-validation and performance metrics.
- Visualization: Scikit-learn integrates with Matplotlib for visualizing data and model results.

Scikit-learn library is focused on modeling the data. Some of the most popular groups of models provided by Sklearn are as follows –

Supervised Learning algorithms: Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.

Unsupervised Learning algorithms: On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

Clustering: This model is used for grouping unlabeled data.

Cross Validation: It is used to check the accuracy of supervised models on unseen data.

Dimensionality Reduction: It is used for reducing the number of attributes in data which can be further used for summarisation, visualisation and feature selection.

Ensemble methods: As name suggest, it is used for combining the predictions of multiple supervised models.

Feature extraction: It is used to extract the features from data to define the attributes in image and text data.

Feature selection: It is used to identify useful attributes to create supervised models.

1.14 Flow Graph of ML Stages

The crucial part of any machine learning project is the workflow behind the project. It serves as an integral tool to determine the success of the project. In this article, we will go over some of the essential aspects involved in an ML workflow. Such as How a standard ML workflow works, the different types of algorithms available, some of the best practices and tools available.

Machine learning project workflow defines the steps involved in executing an ML project.

These steps include:

- Data Collection
- Data Pre-processing
- Building Datasets
- Model Training or Selection
- Model Deployment
- Prediction

- Monitoring Models
- Maintenance, Diagnosis, and Retraining

While the above is a typical machine learning workflow, a lot depends on the project's scope. So have a flexible workflow, start small and scale up to production-ready projects.

The goal of a machine learning workflow is to ensure the project's successful execution. Take a look at the sample **machine learning workflow diagram** below:

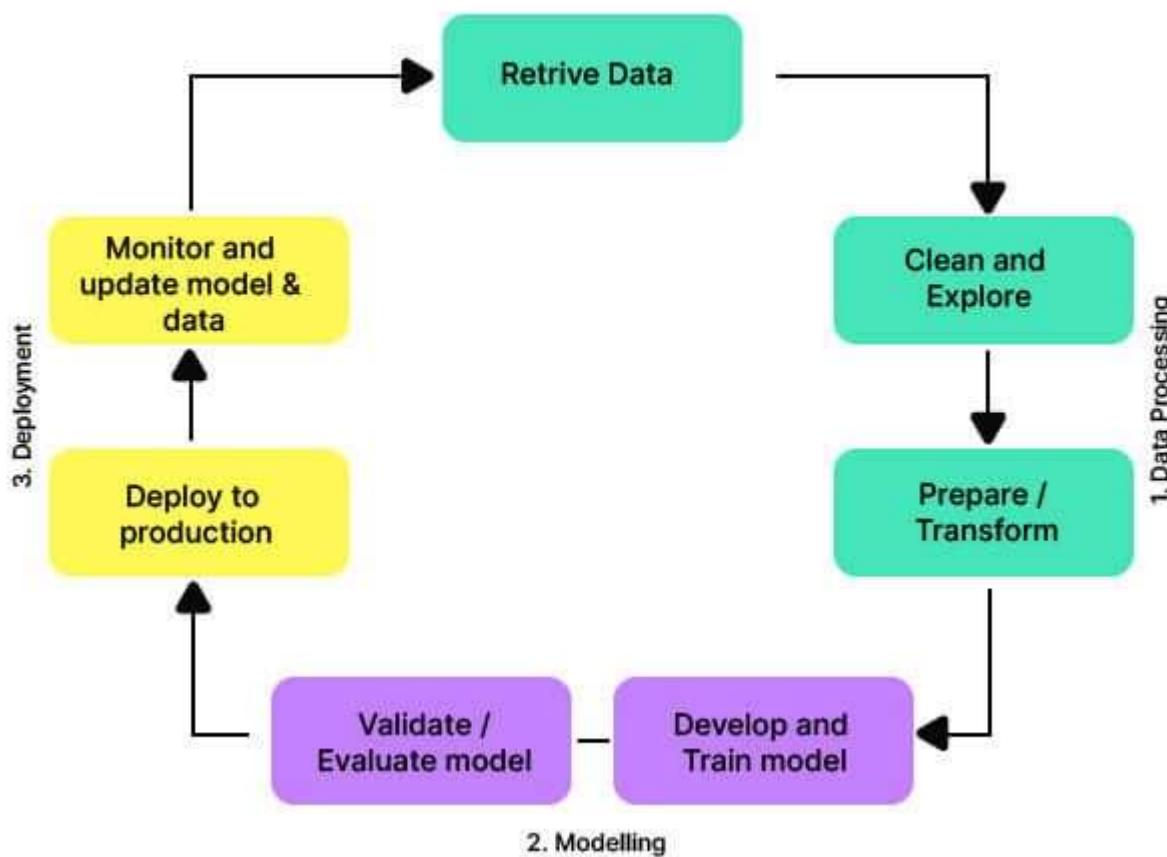


Fig:[ML Flowgraph](#)

Reference: https://d2e931syjhr5o9.cloudfront.net/machine_learning_workflow_diagram_5f04e2c391.jpg

It shows the entire lifecycle of a general machine learning workflow; Next section goes into detail about each machine learning workflow steps.

First stage - Defining the problem:

Before any ML project, the question that needs an answer is:

'What is the problem trying to solve here?', 'Is ML the right approach for this problem?'.

A well-defined problem allows you to choose the right approach to create the solution. But, machine learning depends on data, and there is no answer to how much.

The type of problem you decide to take on also determines the kind of data you need for that particular problem. For example, if you're trying to solve an IoT problem, you may need to work with real-time data.

Second stage - Data collection and preprocessing:

Data collection is the most crucial step in the entire process; the data's quality is the key. You must retrieve, clean, and prepare the data from the source.

Retrieving the data: Data flows into businesses daily, which is in databases or third-party applications. You can also use public data repositories and then pull all the data into one single repository.

Clean the data: Large amounts of data is not equal to clean data. You must clean duplicates, correct errors, input missing data, and structure them. You need to delete garbage data and unwanted noise and eradicate any misinformation.

Prepare the data: After data cleansing, you need to format all those data into a model-ready format. First, you must split your data into train, validation, and test data sets. If required, you might combine many attributes into one and label them.

Post the data collection and cleansing, build the datasets needed for training. Some of the datasets you require are:

Training set: Dataset that enables the model to process information by defining parameters.

Validation set: Verify the model's accuracy and fine-tune the parameters for better results.

Test set: Used to find any bugs or mis-trainings and to test the performance of the model.

Third stage – Training:

Exciting part of the workflow, where develop, train, validate, and test our model. The best practice is to choose an algorithm based on availability of resources such as, computational capabilities, and hardware.

There are several ML algorithms and three different types of machine learning techniques. They are:

Supervised learning:

Feeding labeled data to an algorithm with a set expected outcome. Everything functions in a controlled environment where the deviation is minimal. The accuracy of the result is generally higher. Supervised learning is further divided into classification and regression.

There are several algorithms we can put to use in supervised learning, such as:

- K-Nearest Neighbor
- Naive Bayes
- Decision Trees/Random Forest
- Support Vector Machine
- Logistic Regression
- Linear Regression
- Support Vector Regression
- Decision Trees/Random Forest
- Gaussian Processes Regression
- Ensemble Methods

Unsupervised learning:

This feed unlabelled data to an algorithm, resulting in unexpected outcomes. This is complex and less accurate.

Unsupervised learning is categorized into clustering and association. There are several algorithms that we can put to use as well.

- Gaussian mixtures
- K-Means Clustering
- Boosting
- Hierarchical Clustering
- K-Means Clustering
- Spectral Clustering
- AIS
- SETM
- Apriori
- Latter

Reinforcement learning:

Reinforcing good behaviour and punishing bad behaviour, data labelling is not required. Reinforcement learning is categorized into value-based, policy-based, and model based.

Below are the algorithms used in reinforcement learning.

- Monte Carlo
- Q-learning
- SARSA
- Q-learning - Lambda
- SARSA - Lambda
- DQN - Deep Q Network
- DDPG - Deep Deterministic Policy Gradient
- A3C - Asynchronous Advantage Actor-Critic Algorithm
- NAF - Q-Learning with Normalized Advantage Functions
- TRPO - Trust Region Policy Optimization
- PPO - Proximal Policy Optimization
- TD3 - Twin Delayed Deep Deterministic Policy Gradient
- SAC - Soft Actor-Critic

Based on the problem definition, solution can depend on any of the above algorithms. Train the model using the training parameters for the classification. Then further tune the validation set, and finally test performance using the test set. Some libraries have functions for all algorithms such as TensorFlow, sci-kit-learn, and PyTorch.

Fourth stage – Evaluation:

When we reach a point where test data gives you near accurate results, then need to move on to the evaluation stage. it can identify the best model that provides near-to-accurate results. From here on, ML engineer can retrain, adjust the parameters, or deploy the model into production.

Fifth Stage - Deployment:

Post the evaluation stage; model is generally a proof of concept. Now it need to convert the evidence into an actual viable product. Getting the model into production gives the ability to learn and relearn more.

Some of the ways that can make sure to improve your model in production are:

A/B testing: Comparing the performance of the existing process and system with your current model. You can improve the performance of model based on the result.

Machine learning APIs: The best way to communicate with data sources and other services is through APIs. This is very important if planning to offer your model as a service or product to others.

Detailed documentation: Good documentation goes a long way with any tool or service. How to use your model, what results to expect, and where to access them are some of them.

Sixth and final stage Prediction, Maintenance, Diagnosis, and Retraining:

This is where all the hard work generates fruits. Real-time data feeds into the model and starts sending predictions back. To improve performance, track the model, log errors, run diagnoses and retrain them.

1.15 Practical: Hello World to ML!

First task to build a classifier that distinguishes whether a given input is an ‘Apple’ or ‘Orange’. As seen from the table below.

Weight	Texture	Label
150g	Bumpy	Orange
170g	Bumpy	Orange
140g	Smooth	Apple
130g	Smooth	Apple

There are two main feature that we are going to use, [Weight, Texture] given these two inputs model will predict if it is an Orange or an Apple. Below section discuss each line of code that is used to build the classifier.

We are going to use an open source package called scikit-learn and from scikit-learn we will be using a decision tree to create the classifier for our model.

```
import sklearn
from sklearn import tree
```

Once we do that we write out our training data,

```
import sklearn
from sklearn import
treefeature = [[140,1], [130, 1], [150, 0], [170, 0]]
label = [0, 0, 1, 1]
```

For our feature, we are using '0 = Bumpy', '1 = Smooth' and for our labels, we are using '0 = Apple', '1 = Orange'.

Once we write out our training data, we use the decision tree to build our classifier and then we use 'fit' to train our data.

```
import sklearn
from sklearn import tree
feature      =      [[140,1],           [130,       1],           [150,       0],           [170,       0]]
label = [0, 0, 1, 1]
clf = tree.DecisionTreeClassifier()
clf = clf.fit(feature, labels)
```

After training the given data, we predict if the given input is Apple or Orange, the result will be binary which will print '0' if it is Apple or '1' if it is Orange.

```
import sklearn
from sklearn import tree
feature = [[140,1], [130, 1], [150, 0], [170, 0]]
label = [0, 0, 1, 1]
clf = tree.DecisionTreeClassifier()
clf = clf.fit(feature, labels)
print ((clf.predict([[150, 0]]))[1])
```

The result has output [1] which show's that the given input is an Orange. This was a simple example of building a classifier, as the number of data will increase so will the accuracy of our prediction model.

1.16 ML Problem Types

This section discuss about the most common types of machine learning (ML) problems along with a few examples. Major types of ML problems and useful techniques to attain solution are tabulated below

Problem types	Details	Algorithms
Regression	When the need is to predict numerical values, such kinds of problems are called regression problems. For example, house price prediction	Linear regression, K-NN, random forest, neural networks
Classification	When there is a need to classify the data in different classes, it is called a classification problem. If there are two classes, it is called a binary classification problem. When it is multiple classes, it is multinomial classification. For example, classify whether a person is suffering from a disease or otherwise. Classify whether a stock is “buy”, “sell”, or “hold”.	Logistic regression, <u>random forest</u> , K-NN, gradient boosting classifier, neural networks
Clustering	When there is a need to categorize the data points in similar groupings or clusters, this is called a clustering problem.	<u>K-Means</u> , DBSCAN, Hierarchical clustering, Gaussian mixture models, BIRCH
Time-series forecasting	When there is a need to predict a number based on the time-series data, it is called a time-series forecasting problem. A time series is a sequence of numerical data points in successive order. Time series data means that data is in a series of particular time periods or intervals. For example, a time-series forecasting problem is about forecasting the sales demand for a product, based on a set of input data such as previous sales figures, consumer sentiment, and weather. Another kind of time series problem is demand forecasting.	<u>ARIMA</u> , <u>SARIMA</u> , LSTM, Exponential smoothing, Prophet, GARCH, TBATS, Dynamic linear models
Anomaly detection	When there is a need to find the outliers in the dataset, the problem is called an anomaly detection problem. In other words, if a given record can be classified	IsolationForest, Minimum covariance determinant, Local

	as an outlier or unexpected event/item, this can be called an anomaly detection problem. For example, credit card fraud transactions detection is an anomaly detection problem.	outlier factor, One-class SVM
Ranking	When there is a need to order the results of a request or a query based on some criteria, the problem is ranking problems. We rank the output of query execution based on scores we assign to each output based on some algorithms. These algorithms are called a ranking algorithm. Recommendation engines make use of the ranking algorithm to recommend the next items.	<u>Bipartite ranking</u> (Bipartite Rankboost, Bipartite RankSVM)
Recommendation	When there is a need to recommend such as “next item” to buy or “next video” to watch or “next song” to listen to, the problem is called a recommendation problem. The solutions to such problems are called recommender systems.	Content-based and collaborative filtering machine learning methods
Data generation	When there is a need to generate data such as images, videos, articles, posts, etc, the problem is called a data generation problem.	Generative adversarial network (GAN), Hidden Markov models
Optimization	When there is a need to generate a set of outputs that optimize outcomes related to some objective (objective function), the problem is called an objective function.	Linear programming methods, genetic programming

1.17 Need for Model Evaluation

Evaluating your model is key to make good and accurate predictions. Needs for Model evaluation are listed below:-

- Machine models are built on a subset of the total data, which is called the **training data**, and they are used to predict on new data that is not part of this training subset.

- If a model is totally adapted to its training data, it would fail to predict accurately any new data that was not exactly as in the training set (this phenomenon is known as **overfitting**).
- On the other hand, if your model is too general it would predict poorly on particular cases (**underfitting**).
- A good model should be perfectly balanced to avoid both. By holding out part of the data from the training set and evaluating model with this subset of **test data**, will be able to measure the real performance of your model when a new case appears. Test data, of course, should never be part of the training data for the evaluation to be significant.

With scikit-learn you can measure your model's performance with different evaluation metrics for classification models and for regression models.

1.18 About GitHub

There are many ways you can manage and store your writing projects. Some people prefer cloud storage services (like Dropbox) or online editors (like Google Docs), while others use desktop applications (like Microsoft Word). We are going to use something called GitHub.



Fig: [Git & GitHub](#)

Reference: <https://www.freecodecamp.org/news/git-and-github-the-basics/>

If you've never heard of GitHub, it's the world's most popular destination to store and maintain open-source code. That might sound like a crazy place to host your writing, but it's not! After all, code is just lines and lines of text, like your article, story, or dissertation.

Around 2013, GitHub started encouraging people to create repositories for all kinds of information, not just code. GitHub never really left its coding roots, but some people still use it to store writing and other non-coding projects. For example, one person used Git and GitHub to write an instructional book, while another wrote a novel.

1.19 Concept of Git

Git is an open-source program created by [Linus Torvalds](#), of Linux fame. Git tracks changes to documents and makes it easier for multiple people to work on the same document remotely. In tech-speak, it's called a distributed version control system (or distributed VCS). Git doesn't arbitrarily save versions of your documents at set intervals. Instead, it stores changes to your documents only when you tell it to.

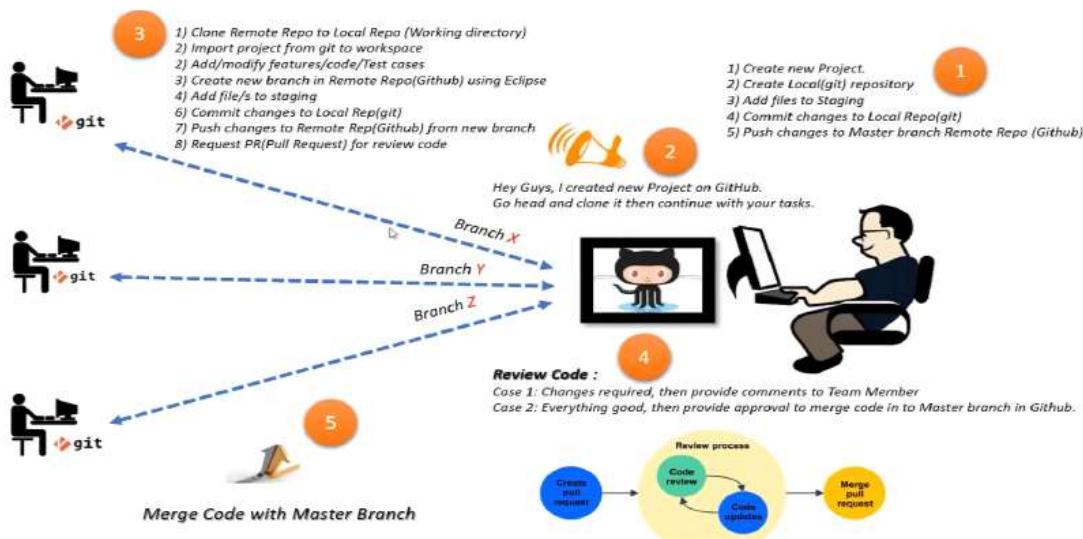


Fig: [Version Control System](#)

Reference: <https://www.youtube.com/watch?v=OqmSzXDrJBk>

1.20 Git Staging

Staging is the process of organizing and preparing our project files for a commit. It is the intermediate step between modifying our files and storing them permanently in the repository. In this tutorial, we will be looking at the Staging Area and try to understand its role.

What is the Staging Area?

- The staging area is an **intermediate step** between making changes to files and capturing the snapshots of these updates. It is sometimes also known as **Git Index**.
- We reach the staging area when we have completed making changes to our files and are ready to commit these changes permanently.

- Files in the working directory are not tracked by Git. Git will only start tracking changes of those files which are added to the staging area. Whenever we try to commit, only the snapshots of those files are captured which were added to the staging area and are stored permanently in the repository.



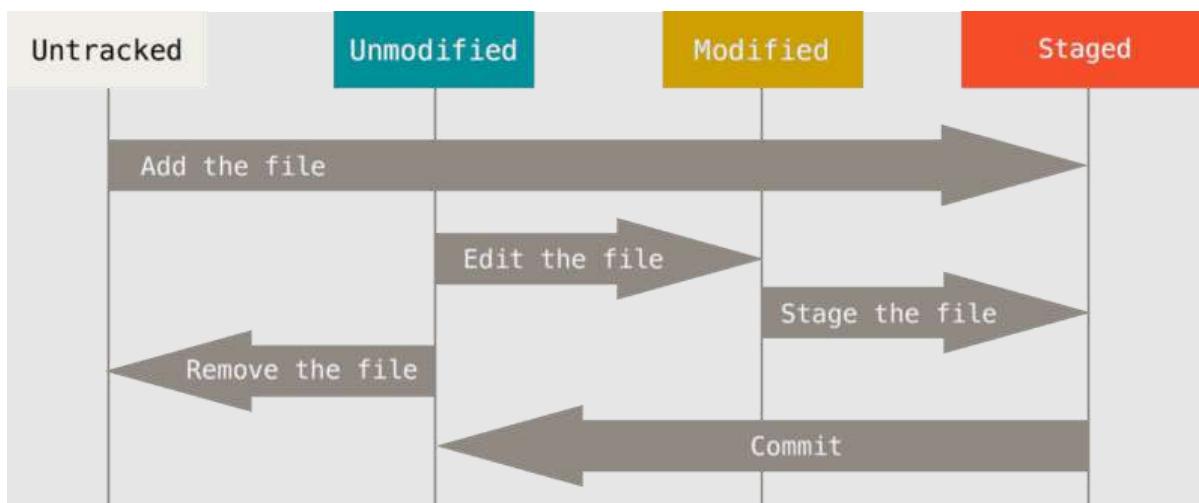
[Fig: Tracking of untracked files](#)

Reference: <https://asanzdiego.github.io/curso-nivelacion-bigdata-2017/recursos/slides/img/git/git-staging-area-bis.png>

Why is the Staging Area needed?

Staging helps in keeping the commits **atomic** which makes it easier to understand the project. We can modify multiple files in the working directory but only add some of them to the staging area and then commit thus making it easier to roll back just a part of the project that involves those files instead of reverting the entire project to a previous version. In this way, staging gives developers more control over how they want to achieve version control.

Tracked & Untracked Stages of Github



[Fig:Intermediate Snapshots of Staging](#)

Reference: <https://asanzdiego.github.io/curso-nivelacion-bigdata-2017/recursos/slides/img/git/git-staging-area-bis.png>

Remember that each file in your working directory can be in one of two states: tracked or untracked. In Github terms “stage” and “snapshot” are analogous to use. Tracked files are files that were in the last stages (any other than tracked stage); they can be unmodified, modified, or staged. Untracked files are everything else – any files in your working directory that were not in your last snapshot and are not in your staging area. When you first clone a repository, all of your files will be tracked and unmodified because you just checked them out and haven’t edited anything.

As you edit files, Git sees them as modified, because you’ve changed them since your last commit. You stage these modified files and then commit all your staged changes, and the cycle repeats.

1.21 Git Commands

With Git, you record local changes to your code using a *command-line* tool, called the “Git Shell” (you can use Git in other command-line tools — Refer to Git Shell through the following sections). Command-line lets you enter commands to view, change, and manage files and folders in a simple terminal, instead of using a graphical user interface (GUI). If you have not used command-line before, don’t worry, once you get started, it is incredibly straightforward.

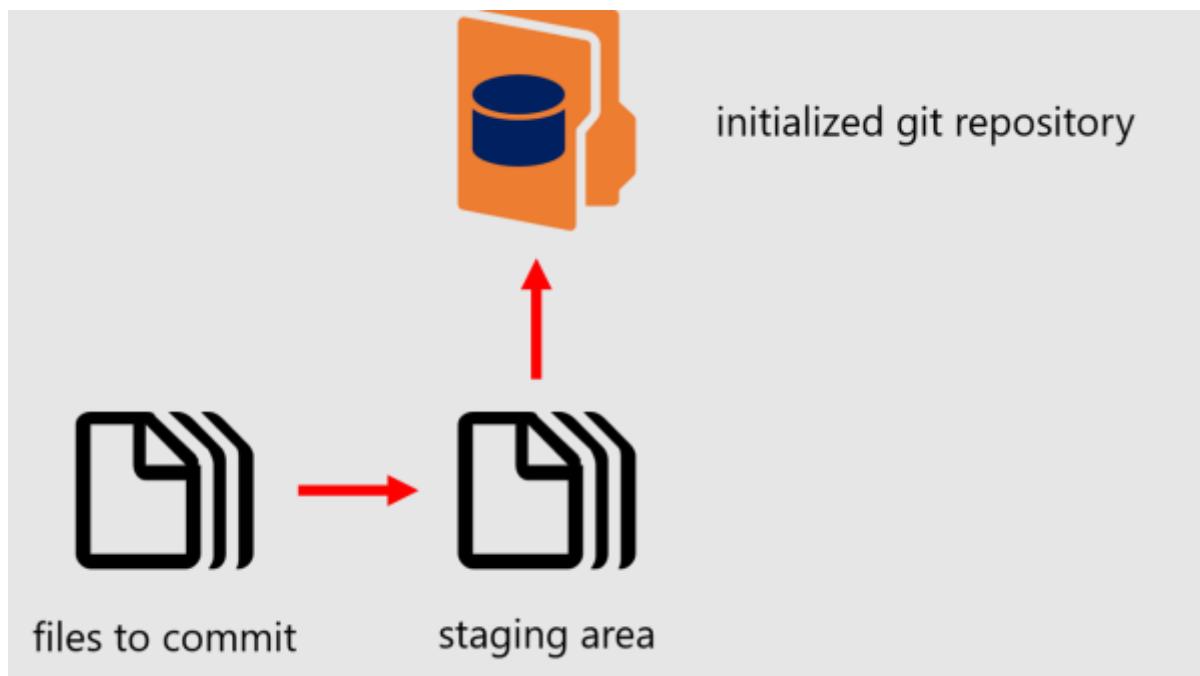


Fig: Git-Hub Staging

Reference: <https://www.nobledesktop.com/learn/git/stage-commit-files>

Essentially, when using Git, you make changes to your code files as you normally would during the development process. When you have completed a coding milestone, or want to snapshot certain changes, you add the files you changed to a staging area and then commit them to the version history of your project (repository) using Git. Below, you'll learn about the Git commands you use for those steps.

Terminal Commands

While using Git on the command line, chances are you will also use some basic terminal commands while going through your project and system files / folders, including:

- **pwd** - check where you are in the current file system
- **ls** - list files in the current directory (folder)
- **cd [directory-name]** - moves to the given directory name or path
- **mkdir [directory-name]** - makes a new directory with the given name

Creating Repositories

When you wish to utilize Git for a project, the first command you must do is *git init*, with the name of your project:

git init [project-name]

You run this command on the Git Shell command-line in the main *directory* (folder) of your project, which you can navigate to in the Shell using the commands listed above. Once you run this command, Git creates a hidden .git file inside the main directory of your project. This file tracks the version history of your project and is what turns the project into a Git *repository*, enabling you to run Git commands on it.

Making Changes

git add [file] or git add *

Once you make changes to your files and choose to snapshot them to your project's version history, you have to add them to the staging area with *git add*, by file name, or by including all of the files in your current folder using *git add **.

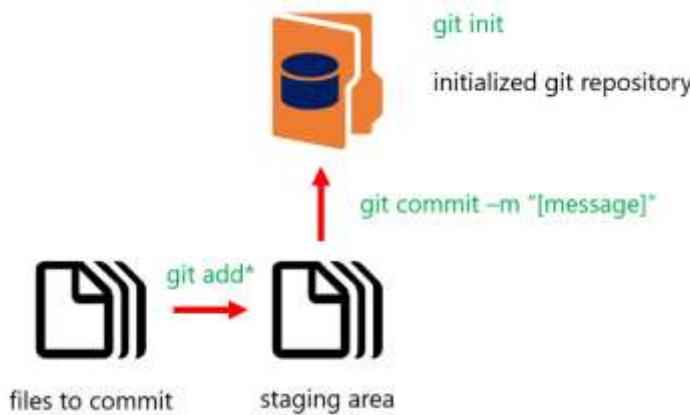
git commit -m “[message]”

To finally commit the changes you made to your files from the staging area to your repository's version history, you need to run *git commit* with a descriptive message of what changes you made.

git status

If at any point, you wish to view a summary of the files you have changed and not yet committed, simply run `git status` in your project's repository on the Git Shell command-line.

How It Works



[GitHub Workflow](#)

Reference: <https://docs.github.com/en/get-started/getting-started-with-git/git-workflows>

Now, with the basic Git commands in place, you can utilize Git to snapshot the version history of your project. Simply initialize a new repository by running `git init` in your project's main directory. Using `git add *`, or `git add` with specific file names, you add your changes to the staging area. Finally, using `git commit`, you can add your changes to the repository's version history.

1.22 Practical: Working with GitHub Environment

- Register / Sign in yourself on Github.

[GitHub: Where the world builds software - GitHub](#)



Sign in to GitHub

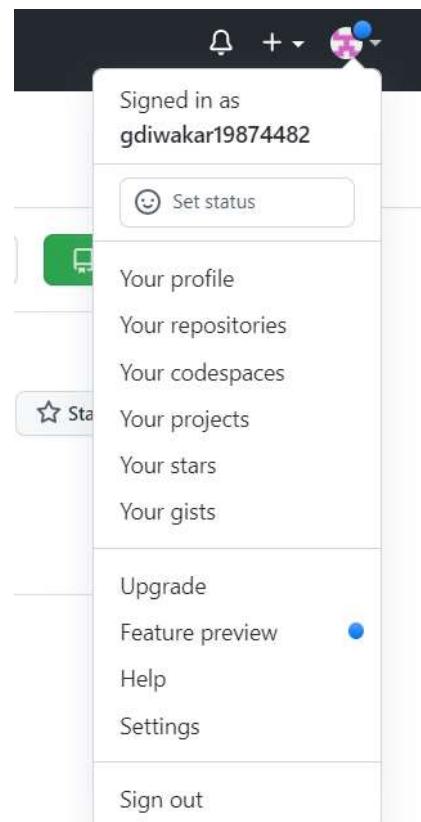
Username or email address

Password [Forgot password?](#)

[Sign in](#)

New to GitHub? [Create an account.](#)

- Switch to Repositories tab on your profile



- Create a new repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *	Repository name *
gdiwakar19874482	/ test2
Great repository names are short and memorable. Need inspiration? How about fuzzy-funicular ?	
Description (optional)	
<input checked="" type="radio"/> Public Anyone on the internet can see this repository. You choose who can commit.	
<input type="radio"/> Private You choose who can see and commit to this repository.	
Initialize this repository with:	
Skip this step if you're importing an existing repository.	
<input checked="" type="checkbox"/> Add a README file <small>This is where you can write a long description for your project. Learn more.</small>	
<input type="checkbox"/> Add .gitignore <small>Choose which files not to track from a list of templates. Learn more.</small>	

- Open Anaconda Command Prompt and install git library

pip install git

- Create a directory with two random .txt files. Place any piece of text in those files.

Configure the access of github profile on local github library

git config --global user.name username

git config --global user.email useremail

** Place your github login username and email information.

- Change the folder to as current working directory. And run git init code

```
(keras-gpu) C:\Git_hub_test>git init
Initialized empty Git repository in C:/Git_hub_test/.git/
```

- Git status shows files are untracked

```
(keras-gpu) C:\Git_hub_test>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file1.txt
    file2.txt

nothing added to commit but untracked files present (use "git add" to track)
```

- Add file1.txt to stage and check git status

```
(keras-gpu) C:\Git_hub_test>git add file1.txt

(keras-gpu) C:\Git_hub_test>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   file1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file2.txt
```

- Now add another file2.txt to staging stage and check git status

```
keras-gpu) C:\Git_hub_test>git add file2.txt

keras-gpu) C:\Git_hub_test>git status
on branch master

 0 commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   file1.txt
    new file:   file2.txt
```

- Commit the traced file using git commit command. You can also mention the commit message.

```
(keras-gpu) C:\Git_hub_test>git commit -m "Test_Github"
[master (root-commit) 46b0a33] Test_Github
 2 files changed, 2 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
```

- Add the address of repository to where you test files are to be uploaded.

gdiwakar19874482 / testset1 · Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master had recent pushes 25 minutes ago

Compare & pull request

main · 2 branches · 0 tags

gdiwakar19874482 Initial commit

README.md · Initial commit

README.md

testset1

About

No desc...

Read...

0 star

1 wat...

0 for...

Releases

No releases Create a ne...

Clone

HTTPS · SSH · GitHub CLI

<https://github.com/gdiwakar19874482/testset1.git>

Use Git or checkout with SVN using the web URL...

Open with GitHub Desktop

Download ZIP

```
(keras-gpu) C:\Git_hub_test>git remote add origin https://github.com/gdiwakar19874482/testset1.git
```

- Upload your local repository to github using git push command

```
(keras-gpu) C:\Git_hub_test>git push origin master
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 285 bytes | 285.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/gdiwakar19874482/testset1/pull/new/master
remote:
To https://github.com/gdiwakar19874482/testset1.git
 * [new branch]      master -> master

(keras-gpu) C:\Git_hub_test>git remote -v
origin  https://github.com/gdiwakar19874482/testset1.git (fetch)
origin  https://github.com/gdiwakar19874482/testset1.git (push)
```

****Note:** GUI window will request for read/ write access.

- You will observe and conclude that your local repository gets uploaded to github repository

The screenshot shows a GitHub repository page for 'gdiwakar19874482 / testset1'. The 'Code' tab is active. At the top, there's a yellow banner stating 'master had recent pushes 30 minutes ago'. Below it, there are buttons for 'Compare & pull request', 'Go to file', 'Add file', and 'Code'. The main area shows the commit history: 'This branch is 1 commit ahead, 1 commit behind main'. It lists two commits from 'gdiwakar19874482 Test_Github' made 33 minutes ago, each adding 'file1.txt' and 'file2.txt' to the 'Test_Github' directory. On the right side, there are sections for 'About' (no description, website, or bio), 'Releases' (no releases published), and a button to 'Create a new release'. At the bottom, there's a suggestion to 'Help people interested in this repository understand your project by adding a README.' with a 'Add a README' button.

Unit 2: Supervised Machine Learning

Learning Outcomes:

- Understand application-oriented difference between regression and classification
- Understand concept and working of major Supervised machine learning algorithms.
- Implement and exhibit the performance of major Supervised machine learning algorithms on case-study datasets.
- Develop solution for generalizing a model and handle bias-variance tradeoff

2.1 Linear Regression: Concept

2.1.1 Regression:

Regression is a process of finding the correlations between dependent and independent variables. It helps in predicting the continuous variables such as prediction of Market Trends, prediction of House prices, etc.

The task of the Regression algorithm is to find the mapping function to map the input variable (x) to the continuous output variable (y).

Example: Suppose we want to do weather forecasting, so for this, we will use the Regression algorithm. In weather prediction, the model is trained on the past data, and once the training is completed, it can easily predict the weather for future days.

2.1.2 Types of Regression Algorithm:

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression

Before starting the linear regression algorithm, let us get started with least square method which is used as a backbone of regression analysis.

Least Square Method

The least-squares regression method is a technique commonly used in Regression Analysis. It is a mathematical method used to find the best fit line that represents the relationship between an independent and dependent variable.

Line of Best Fit

Line of best fit is drawn to represent the relationship between 2 or more variables. To be more specific, the best fit line is drawn across a scatter plot of data points in order to represent a relationship between those data points.

Regression analysis makes use of mathematical methods such as least squares to obtain a definite relationship between the predictor variable (s) and the target variable. The least-squares method is one of the most effective ways used to draw the line of best fit. It is based on the idea that the square of the errors obtained must be minimized to the most possible extent and hence the name least squares method.

Least Squares Regression Example

Consider an example. Tom who is the owner of a retail shop, found the price of different T-shirts vs the number of T-shirts sold at his shop over a period of one week.

He tabulated this like:

Price of T-shirts in dollars (x)	Number of T-shirts sold (y)
2	4
3	5
5	7
7	10
9	15

Let us use the concept of least squares regression to find the line of best fit for the above data.

Step 1: Calculate the slope 'm' by using the following formula, where n is the number of observations:

$$m = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$

After you substitute the respective values, $m = 1.518$ approximately.

Step 2: Compute the y-intercept value

$$c = \frac{(\sum y \sum x^2) - \sum x \sum xy}{n(\sum x^2) - (\sum x)^2}$$

After you substitute the respective values, $c = 0.305$ approximately.

Step 3: Substitute the values in the final equation

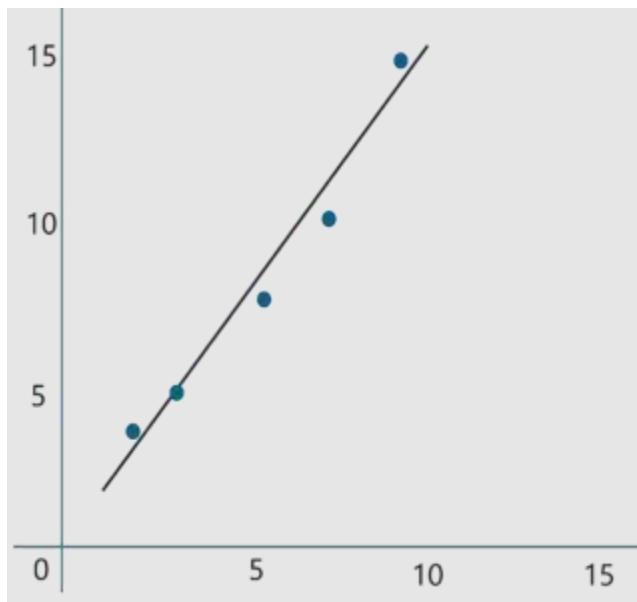
$$y = mx + c$$

Once you substitute the values, it should look something like this:

Table: Tshirt price

Price of T-shirts in dollars (x)	# of T-shirts sold (y)	$Y = mx + c$	Error (Y-y)
2	4	3.341	-0.659
3	5	4.859	-0.141
5	7	7.895	0.895
7	10	10.931	0.931
9	15	13.967	-1.033

Let's construct a graph that represents the $y=mx + c$ line of best fit:



Now, Tom can use the above equation to estimate how many T-shirts of price \$8 can he sell at the retail shop.

$$y = 1.518 \times 8 + 0.305 = 12.45 \text{ T-shirts}$$

The least squares regression method works by minimizing the sum of the square of the errors as small as possible, hence the name least squares. Basically, the distance between the line of best fit and the error must be minimized as much as possible.

A few things to keep in mind before implementing the least squares regression method is:

- The data must be free of outliers because they might lead to a biased and wrongful line of best fit.
- The line of best fit can be drawn iteratively until you get a line with the minimum possible squares of errors.
- This method works well even with non-linear data.
- Technically, the difference between the actual value of 'y' and the predicted value of 'y' is called the Residual (denotes the error).

2.1.3 Linear Regression

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables.

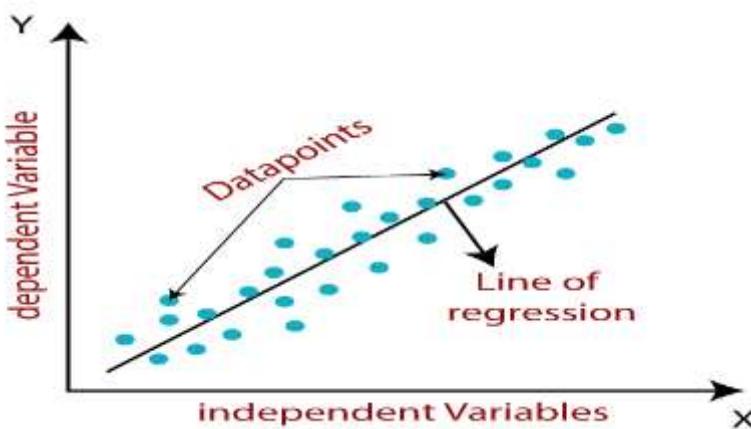


Fig: Relationship between the variables in linear regression
 Reference:<https://www.javatpoint.com/linear-regression-in-machine-learning>

Mathematically, we can represent a linear regression as:

$$y = a_0 + a_1 x + \epsilon$$

Here,

y = Dependent Variable (Target Variable)

x = Independent Variable (predictor Variable)

a_0 = intercept of the line (Gives an additional degree of freedom)

a_1 = Linear regression coefficient (scale factor to each input value).

ϵ = random error

The values for x and y variables are training datasets for Linear Regression model representation.

Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

Simple Linear Regression:

If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

Multiple Linear regression:

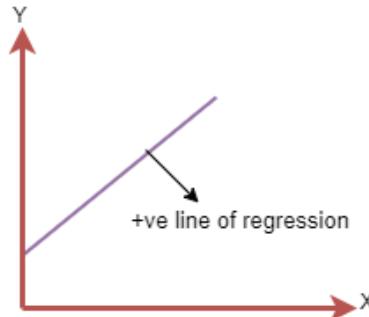
If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Linear Regression Line

A linear line showing the relationship between the dependent and independent variables is called a regression line. A regression line can show two types of relationship:

1. Positive Linear Relationship:

If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.



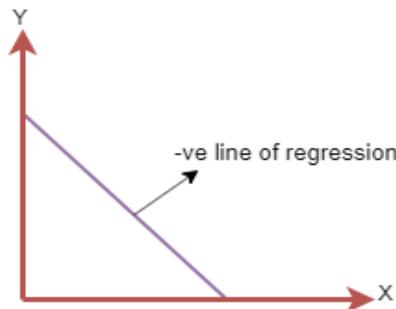
The line equation will be: $Y = a_0 + a_1x$

Fig:Positive linear relationship

Reference: <https://www.javatpoint.com/linear-regression-in-machine-learning>

2. Negative Linear Relationship:

If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.



The line of equation will be: $Y = -a_0 + a_1x$

Fig:Negative linear relationship

Reference: <https://www.javatpoint.com/linear-regression-in-machine-learning>

2.1.4 Assumptions of Linear Regression

These are some formal checks while building a Linear Regression model, which ensures to get the best possible result from the given dataset. Linear relationship between the features and target. Linear regression assumes the linear relationship between the dependent and independent variables. Small or no multicollinearity between the features

Multicollinearity means high-correlation between the independent variables. Due to multicollinearity, it may be difficult to find the true relationship between the predictors and target variables. Or we can say, it is difficult to determine which predictor variable is affecting the target variable and which is not. So, the model assumes either little or no multicollinearity between the features or independent variables.

Homoscedasticity Assumption

Homoscedasticity is a situation when the error term is the same for all the values of independent variables. With homoscedasticity, there should be no clear pattern distribution of data in the scatter plot.

Normal distribution of error terms:

Linear regression assumes that the error term should follow the normal distribution pattern. If error terms are not normally distributed, then confidence intervals will become either too wide or too narrow, which may cause difficulties in finding coefficients.

It can be checked using the q-q plot. If the plot shows a straight line without any deviation, which means the error is normally distributed.

No autocorrelations:

The linear **regression** model assumes no autocorrelation in error terms. If there will be any correlation in the error term, then it will drastically reduce the accuracy of the model. Autocorrelation usually occurs if there is a dependency between residual errors.

Mathematical Intuition:

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

The different values for weights or the coefficient of lines (a_0, a_1) gives a different line of regression, so we need to calculate the best values for a_0 and a_1 to find the best fit line, so to calculate this we use cost function.

Cost function:

The different values for weights or coefficient of lines (a_0, a_1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line. Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.

We can use the cost function to find the accuracy of the mapping function, which maps the input variable to the output variable. This mapping function is also known as Hypothesis function.

For Linear Regression, we use the Mean Squared Error (MSE) cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:

For the above linear equation, MSE can be calculated as:

$$MSE = \frac{1}{N} \sum_{i=1}^n (y_i - (a_1 x_i + a_0))^2$$

where,

N=Total number of observations

y_i = Actual value

$(a_1 x_i + a_0)$ = Predicted value

Residuals:

The distance between the actual value and predicted values is called residual. If the observed points are far from the regression line, then the residual will be high, and so cost function will high. If the scatter points are close to the regression line, then the residual will be small and hence the cost function.

Gradient Descent:

- Gradient descent is used to minimize the MSE by calculating the gradient of the cost function.
- A regression model uses gradient descent to update the coefficients of the line by reducing the cost function.

- It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

2.2 Evaluation Metrics: Regression

The Goodness of fit determines how the line of regression fits the set of observations. The process of finding the best model out of various models is called optimization. It can be achieved by the following method:

R-squared method:

- R-squared is a statistical method that determines the goodness of fit. It measures the strength of the relationship between the dependent and independent variables on a scale of 0-100%.
- The high value of R-square determines the less difference between the predicted values and actual values and hence represents a good model.
- It is also called a coefficient of determination, or coefficient of multiple determination for multiple regression.

It can be calculated from the below formula:

$$R - \text{square} = \frac{\text{Explained Variation}}{\text{Total Variation}}$$

Ordinary Least Square Method:

Ordinary least squares, or linear least squares, estimates the parameters in a regression model by minimizing the sum of the squared residuals. This method draws a line through the data points that minimizes the sum of the squared differences between the observed values and the corresponding fitted values.

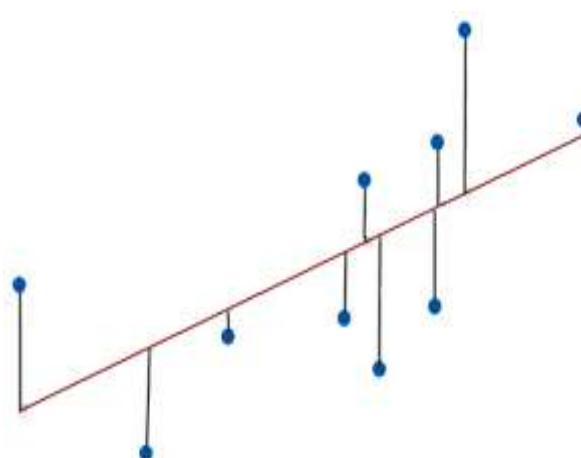


Fig: Ordinary least squares line

Reference: <https://statisticsbyjim.com/glossary/ordinary-least-squares/#:~:text=Ordinary%20least%20squares%2C%20or%20linear,an%20the%20corresponding%20fitted%20values>

- The Ordinary Least Squares procedure seeks to minimize the sum of the squared residuals. This means that given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that ordinary least squares seeks to minimize.
- This approach treats the data as a matrix and uses linear algebra operations to estimate the optimal values for the coefficients. It means that all of the data must be available and you must have enough memory to fit the data and perform matrix operations.
- Ordinary Least Squares is a form of statistical regression used as a way to predict unknown values from an existing set of data. An example of a scenario in which one may use Ordinary Least Squares, or OLS, is in predicting shoe size from a data set that includes height and shoe size. Given the data, one can use the ordinary least squares formula to create a rate of change and predict shoe size, given a subject's height. In short, OLS takes an input, the independent variable, and produces an output, the dependent variable.

OLS method equation:

$$m = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

$$b = \bar{y} - m * \bar{x}$$

x = independent variables

\bar{x} = average of independent variables

y = dependent variables

\bar{y} = average of dependent variables

Ordinary Least Squares method works for both univariate dataset which means single independent variables and single dependent variables and multi-variate dataset which contains a single independent variable set and multiple dependent variables sets.

Adjusted R-Square:

First it needs to understand the demerits of R-Square.

When we fit linear regression models we often calculate the **R-squared** value of the model. The R-squared value is the proportion of the variance in the response variable that can be explained by the predictor variables in the model.

The value for R-squared can range from 0 to 1 where:

- A value of **0** indicates that the response variable cannot be explained by the predictor variables at all.

2. A value of **1** indicates that the response variable can be perfectly explained by the predictor variables.

Although this metric is commonly used to assess how well a regression model fits a dataset, it has one serious drawback:

The drawback of R-squared:

"R-squared will always increase when a new predictor variable is added to the regression model."

Even if a new predictor variable is almost completely unrelated to the response variable, the R-squared value of the model will increase, if only by a small amount.

For this reason, it's possible that a regression model with a large number of predictor variables has a high R-squared value, even if the model doesn't fit the data well.

Fortunately, there is an alternative to R-squared known as **adjusted R-squared**.

The **adjusted R-squared** is a modified version of R-squared that adjusts for the number of predictors in a regression model. It is calculated as:

$$\text{Adjusted } R^2 = 1 - [(1-R^2) * (n-1)/(n-k-1)]$$

where:

- **R²**: The R² of the model
- **n**: The number of observations
- **k**: The number of predictor variables

Because R-squared always increases as you add more predictors to a model, the adjusted R-squared can tell you how useful a model is, *adjusted for the number of predictors in a model*.

The advantage of Adjusted R-squared: Adjusted R-squared tells us how well a set of predictor variables is able to explain the variation in the response variable, *adjusted for the number of predictors in a model*. Because of the way it's calculated, adjusted R-squared can be used to compare the fit of regression models with different numbers of predictor variables.

2.3 Practical: Linear Regression using Python – Scikit Learn library

Let us see how to build a simple linear machine learning model for the Diabetes dataset. This dataset consists of Ten baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements were obtained for each of $n = 442$ diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after baseline. We will apply Linear Regression for this considering one input feature and output target.

The Link for Project is given below:

<https://github.com/Code-Unnati/Module-III/blob/master/Machine%20Learning%20Models/Linear%20Regression.ipynb>

2.4 Bias-Variance Tradeoff

To evaluate the performance of a model on a dataset, we need to measure how well the model predictions match the observed data. For regression models, the most commonly used metric is the mean squared error (MSE), which is calculated as:

$$\text{MSE} = (1/n) * \sum (y_i - f(x_i))^2$$

where:

1. **n**: Total number of observations
2. **y_i**: The response value of the i^{th} observation
3. **f(x_i)**: The predicted response value of the i^{th} observation

The closer the model predictions are to the observations, the smaller the MSE will be. However, we only care about **test MSE** – the MSE when our model is applied to unseen data. This is because we only care about how the model will perform on unseen data, not existing data.

For example, it's nice if a model that predicts stock market prices has a low MSE on historical data, but we *really* want to be able to use the model to accurately forecast future data.

It turns out that the test MSE can always be decomposed into two parts:

1. **The variance**: Refers to the amount by which our function f would change if we estimated it using a different training set.

2. The bias: Refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model.

Written in mathematical terms:

$$\text{Test MSE} = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\epsilon)$$

$$\text{Test MSE} = \text{Variance} + \text{Bias}^2 + \text{Irreducible error}$$

The third term, the irreducible error, is the error that cannot be reduced by any model simply because there always exists *some* noise in the relationship between the set of explanatory variables and the response variable.

Models that have **high bias** tend to have **low variance**. For example, linear regression models tend to have high bias (assumes a simple linear relationship between explanatory variables and response variable) and low variance (model estimates won't change much from one sample to the next).

However, models that have **low bias** tend to have **high variance**. For example, complex non-linear models tend to have low bias (does not assume a certain relationship between explanatory variables and response variable) with high variance (model estimates can change a lot from one training sample to the next).

The Bias-Variance Tradeoff:

The **bias-variance tradeoff** refers to the tradeoff that takes place when we choose to lower bias which typically increases variance, or lower variance which typically increases bias.

The following chart offers a way to visualize this tradeoff:

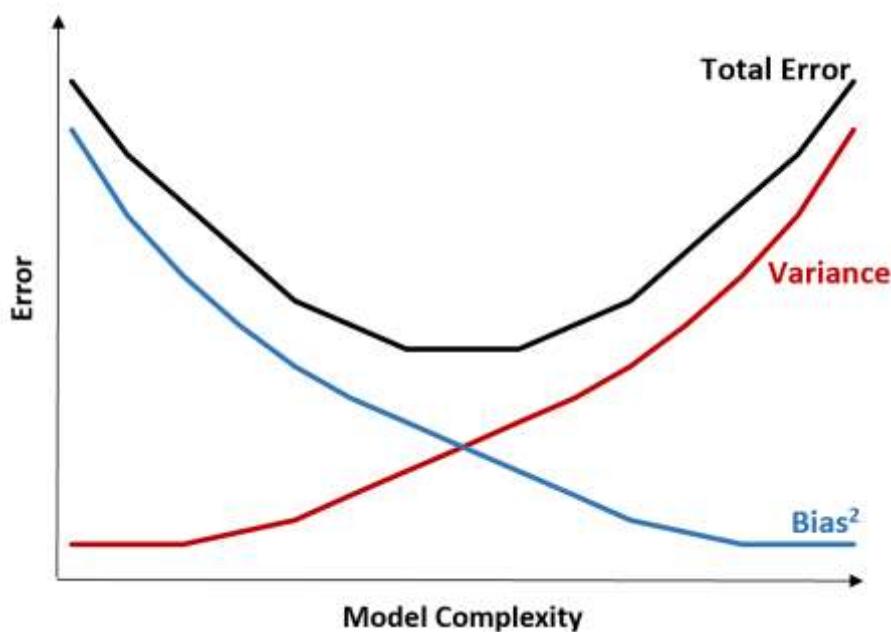


Fig:[Bias Variance Tradeoff](#)

Reference: https://www.statology.org/wp-content/uploads/2020/10/bias_variance1-768x558.png

The total error decreases as the complexity of a model increases but only up to a certain point. Past a certain point, variance begins to increase and total error also begins to increase.

2.5 Handle Bias Variance Tradeoff: Lasso & Ridge Regression

In ordinary multiple linear regression, we use a set of p predictor variables and a response variable to fit a model of the form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \epsilon$$

The values for $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ are chosen using the least square method, which minimizes the sum of squared residuals (RSS):

$$RSS = \sum (y_i - \hat{y}_i)^2$$

where:

- Σ : A symbol that means “sum”
- y_i : The actual response value for the i^{th} observation
- \hat{y}_i : The predicted response value for the i^{th} observation

One problem that often occurs in practice with multiple linear regression is multicollinearity – when two or more predictor variables are highly correlated to each other, such that they do not provide unique or independent information in the regression model. This can cause the coefficient estimates of the model to be unreliable and have high variance. That is, when the model is applied to a new set of data it hasn't seen before, it's likely to perform poorly.

Avoiding Multicollinearity: Ridge & Lasso Regression

Two methods we can use to get around this issue of multicollinearity are **ridge regression** and **lasso regression**.

Ridge regression seeks to minimize the following:

- $RSS + \lambda \sum \beta_j^2$

Lasso regression seeks to minimize the following:

- $RSS + \lambda \sum |\beta_j|$

In both equations, the second term is known as a *shrinkage penalty*.

When $\lambda = 0$, this penalty term has no effect and both ridge regression and lasso regression produce the same coefficient estimates as least squares.

However, as λ approaches infinity the shrinkage penalty becomes more influential and the predictor variables that aren't importable in the model get shrunk towards zero. Below graph can notice impact of regularisation in Lasso & Ridge models.

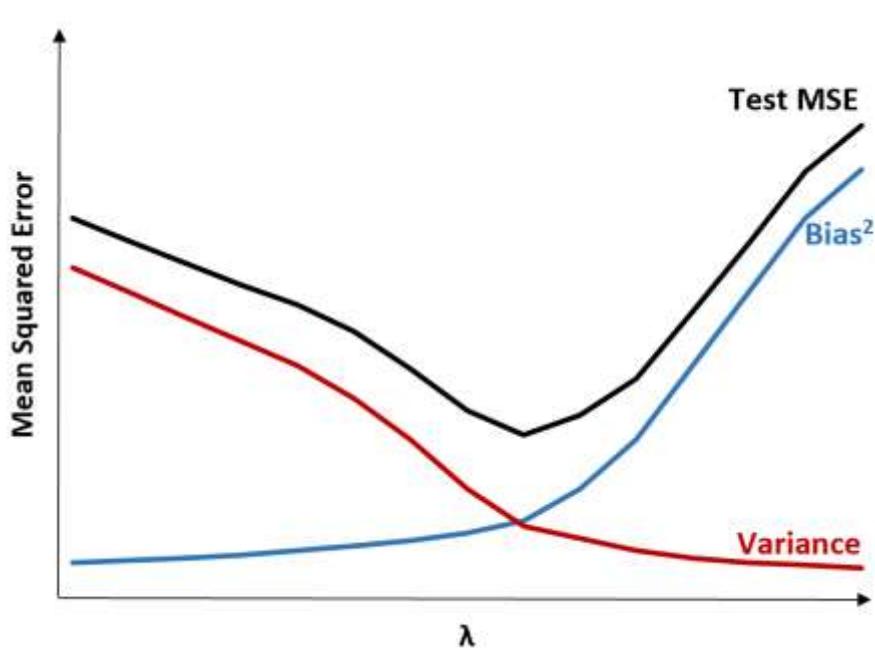


Fig:[Regularised Bias Variance Tradeoff](#)

Reference: <https://www.statology.org/wp-content/uploads/2020/11/ridge1-768x561.png>

2.6 Practical: Implementation of Ridge & Lasso Regression Using Python-Scikit Learn Library

The Link for Project is given below:

2.7 Logistic Regression – Concept

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

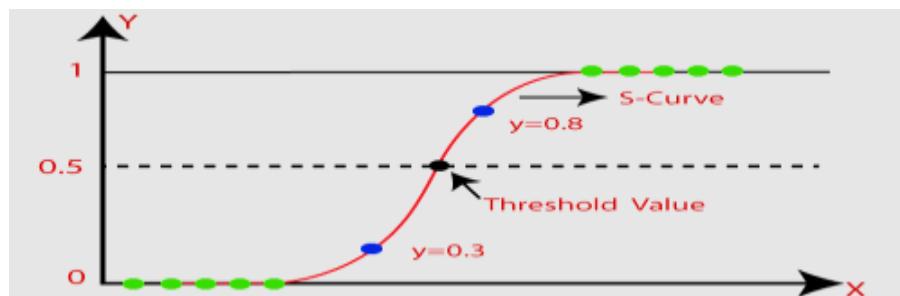


Fig: logistic function
 Reference:<https://www.javatpoint.com/logistic-regression-in-machine-learning>

Logistic Function (Sigmoid Function)

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Assumptions for Logistic Regression

- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.

Logistic Regression Equation

The Logistic regression equation can be obtained from the Linear Regression equation.

- We know the equation of the straight line can be written as:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}; 0 \text{ for } y=0, \text{ and infinity for } y=1$$

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

Types of Logistic Regression

On the basis of the categories, Logistic Regression can be classified into three types:

Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

2.8 Evaluation Metrics: Classification

Classification Accuracy:

The simplest metric for model evaluation is Accuracy. It is the ratio of the number of correct predictions to the total number of predictions made for a dataset.

$$\text{Accuracy} = \frac{\text{Number of Correct predictions}}{\text{Total number of predictions made}}$$

Accuracy is useful when the target class is well balanced but is not a good choice with unbalanced classes.

For example, A dataset with two target classes containing 100 samples. 98 samples belong to class A and 2 samples belong to class B in our training data, our model would give us 98% accuracy. That's why we need to look at more metrics to get a better result.

Logarithmic Loss or Log Loss:

Log Loss can be used when the output of the classifier is a numeric probability instead of a class label. Log loss measures the unpredictability of the extra noise that comes from using a predictor as opposed to the true labels.

For multi-class classification:

$$\text{LogarithmicLoss} = \frac{-1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} * \log(p_{ij})$$

Consider, N samples belong to the M class. where,

y_ij indicates whether sample i belongs to class j or not

p_ij indicates the probability of sample i belonging to class j

Confusion Matrix:

A confusion matrix or error matrix is a table that shows the number of correct and incorrect predictions made by the model compared with the actual classifications in the test set or what type of errors are being made.

This matrix describes the performance of a classification model on test data for which true values are known. It is a n*n matrix, where n is the number of classes. This matrix can be generated after making predictions on the test data.

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	TP True Positive	FP False Positive
	negatives	FN False Negative	TN True Negative

Fig: [alearningaday](#)Reference: <https://alearningaday.blog/2016/09/14/confusion-matrix/>

Here, columns represent the count of actual classifications in the test data while rows represent the count of predicted classifications made by the model.

Let's take an example of a classification problem where we are predicting whether a person is having diabetes or not. Let's give a label to our target variable:

1: A person is having diabetes | 0: A person is not having diabetes

Four possible outcomes could occur while performing classification predictions:

True Positives (TP): Number of outcomes that are actually positive and are predicted positive. For example: In this case, a person is actually having diabetes(1) and the model predicted that the person has diabetes(1).

True Negatives (TN): Number of outcomes that are actually negative and are predicted negative. For example: In this case, a person actually doesn't have diabetes(0) and the model predicted that the person doesn't have diabetes(0).

False Positives (FP): Number of outcomes that are actually negative but predicted positive. These errors are also called **Type 1 Errors**. For example: In this case, a person actually doesn't have diabetes(0) but the model predicted that the person has diabetes(1).

False Negatives (FN): Number of outcomes that are actually positive but predicted negative. These errors are also called **Type 2 Errors**.

2.9 Practical: Logistic Regression using Python-Scikit learn library

Let us consider dataset on our own. We can create linear separable dataset using make_classification class in Sklearn. Then Apply Logistic Regression to build a machine Learning model.

The Link for project is given below:

<https://github.com/Code-Unnati/Module-III/blob/master/Machine%20Learning%20Models/Logistic%20Regression.ipynb>

2.10 Cross Validation: Concept

Cross validation is a technique used in machine learning to evaluate the performance of a model on unseen data. It involves dividing the available data into multiple folds or subsets, using one of these folds as a validation set, and training the model on the remaining folds. This process is repeated multiple times, each time using a different fold as the validation set. Finally, the results from each validation step are averaged to produce a more robust estimate of the model's performance.

The main purpose of cross validation is to prevent overfitting, which occurs when a model is trained too well on the training data and performs poorly on new, unseen data. By evaluating the model on multiple validation sets, cross validation provides a more realistic estimate of the model's generalization performance, i.e., its ability to perform well on new, unseen data.

There are several types of cross validation techniques, including k-fold cross validation, leave-one-out cross validation, and stratified cross validation. The choice of technique depends on the size and nature of the data, as well as the specific requirements of the modelling problem.

Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set. The three steps involved in cross-validation are as follows :

- Reserve some portion of sample data-set.
- Using the rest data-set train the model.
- Test the model using the reserve portion of the data-set.

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



Fig:[Cross-Validation](#)

Reference: <https://media.geeksforgeeks.org/wp-content/uploads/crossValidation.jpg>

2.11 K-Nearest Neighbors – Concept

2.11.1 Distance Measures

Distance metrics play an important role in machine learning. They provide a strong foundation for several machine learning algorithms like k-nearest neighbors for supervised learning and k-means clustering for unsupervised learning. Different distance metrics are chosen depending upon the type of the data. So, it is important to know the various distance metrics and the intuitions behind it.

An effective distance metric improves the performance of our machine learning model, whether that's for classification tasks or clustering.

There are several measures of distance that can be used, and it is important to be aware of them while considering the best solution for a given situation to avoid errors and interpretation issues.

Types of Distance Metrics in Machine Learning

- Euclidean Distance
- Euclidean Distance represents the shortest distance between two points.
- Euclidean distance formula can be used to calculate the distance between two data points in a plane.

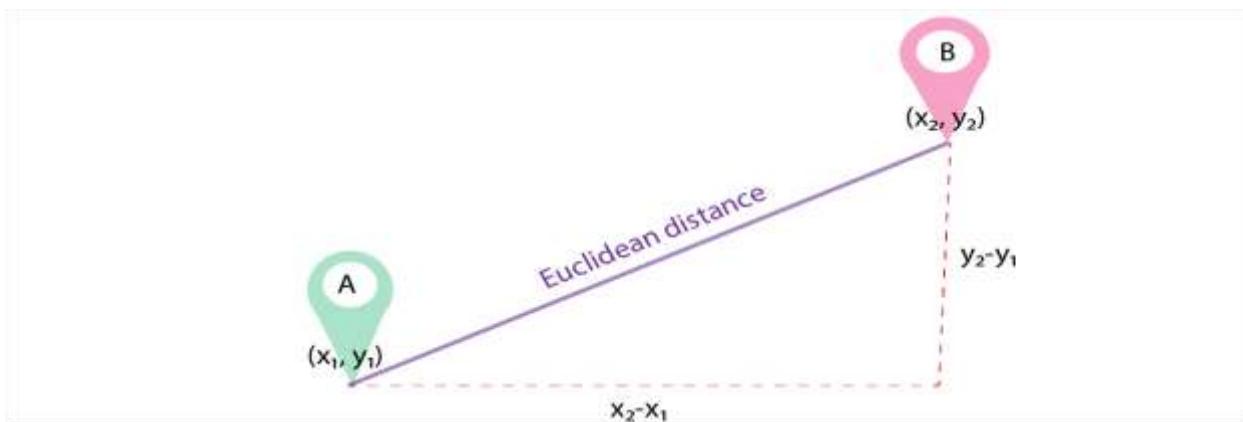


Fig: Euclidean Distance

Reference:<https://medium.com/analytics-vidhya/role-of-distance-metrics-in-machine-learning-e43391a6bf2e>

- Euclidean distance is generally used when calculating the distance between two rows of data that have numerical values, such as floating point or integer values.
- If columns have values with differing scales, it should be normalized or standardized before calculating the Euclidean distance. Otherwise, columns that have large values will dominate the distance measure.
- Euclidean distance is calculated as the square root of the sum of the squared differences between the two vectors.

$$D_e = \left(\sum_{i=1}^n (p_i - q_i)^2 \right)^{1/2}$$

where,

n = number of dimensions

p_i, q_i = data points

Manhattan Distance

Manhattan Distance is the sum of absolute differences between points across all the dimensions. We use Manhattan distance, also known as city block distance, or taxicab geometry if we need to calculate the distance between two data points in a grid-like path just like a chessboard or city blocks. The name taxicab refers to the intuition for what the measure calculates: the shortest path that a taxicab would take between city blocks (coordinates on the grid).

Let's say, we want to calculate the distance, d , between two data points- A and B.

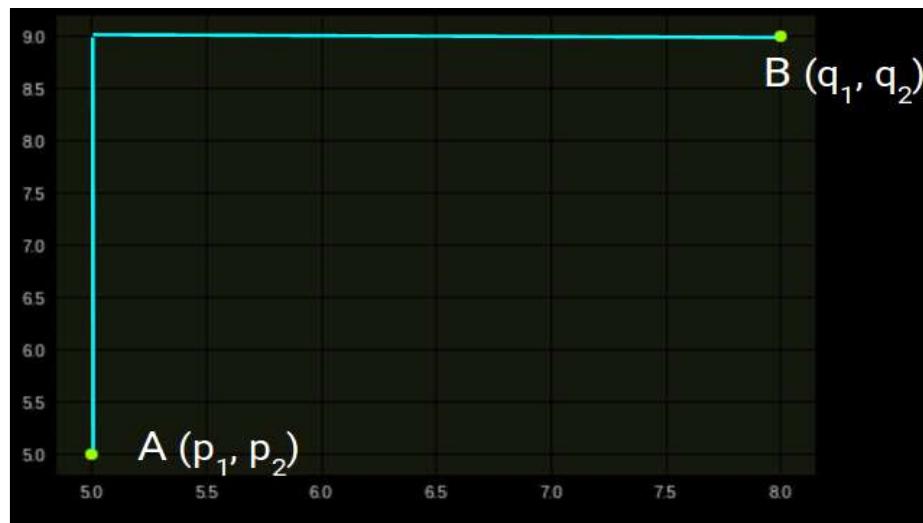


Fig: Euclidean Distance between A and B

Reference: <https://medium.com/analytics-vidhya/role-of-distance-metrics-in-machine-learning-e43391a6bf2e>

Distance d will be calculated using an absolute sum of difference between its cartesian co-ordinates as below:

$$d = |p_1 - q_1| + |p_2 - q_2|$$

And the generalized formula for an n -dimensional space is given as:

$$D_m = \sum_{i=1}^n |p_i - q_i|$$

where,

n = number of dimensions

p_i, q_i = data points

The Manhattan Distance is preferred over the Euclidean distance metric as the dimension of the data increases. This occurs due to something known as the 'curse of dimensionality'.

Minkowski Distance

Minkowski Distance is the generalized form of Euclidean and Manhattan Distance. Minkowski Distance calculates the distance between two points. It is a generalization of the Euclidean and Manhattan distance measures and adds a parameter, called the “order” or “p”, that allows different distance measures to be calculated. The Minkowski distance measure is calculated as follows:

$$D = \left(\sum_{i=1}^n |p_i - q_i|^p \right)^{1/p}$$

where “p” is the order parameter.

When p is set to 1, the calculation is the same as the Manhattan distance. When p is set to 2, it is the same as the Euclidean distance.

- p=1: Manhattan distance
- p=2: Euclidean distance

Intermediate values provide a controlled balance between the two measures.

- It is common to use Minkowski distance when implementing a machine learning algorithm that uses distance measures as it gives control over the type of distance measure used for real-valued vectors via a hyperparameter “p” that can be tuned.

Hamming Distance

Hamming Distance measures the similarity between two strings of the same length. The Hamming Distance between two strings of the same length is the number of positions at which the corresponding characters are different.

$$d = \min \{ d(x, y) : x, y \in C, x \neq y \}$$

So, this is how we can calculate the distance between datapoints, which is the core concept behind our next algorithm called K-Nearest Neighbors.

2.11.2 Geometric Intuition of K-NN

KNN assumes that all our data points are geometrically close to each other or in other words the neighbourhood points should be close to each other.

K-Nearest Neighbor (K-NN) Algorithm for Machine Learning

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears, it can be easily classified into a well suited category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs' images and based on the most similar features it will put it in either cat or dog category.



Fig: KNN Classifier

Reference: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>

2.11.3 Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x_1 , so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

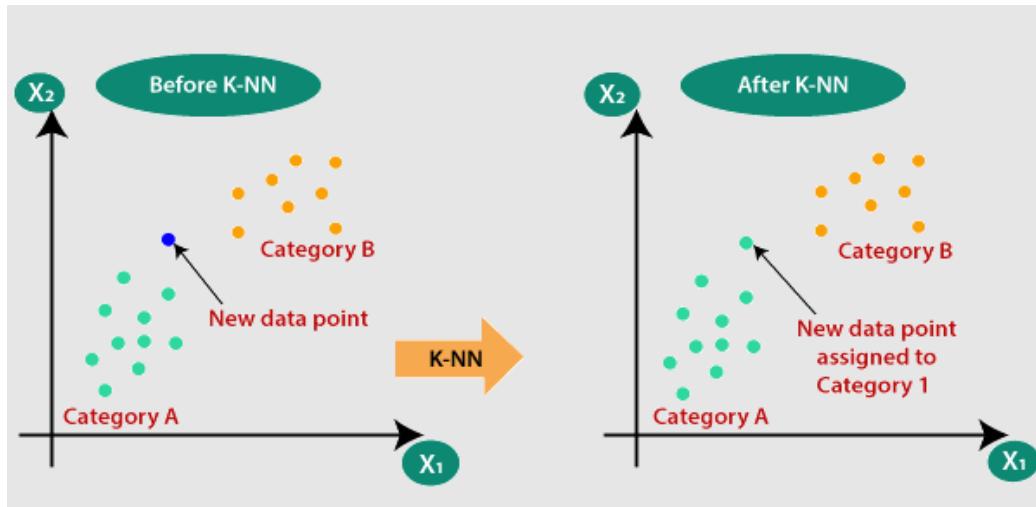


Fig: Before and After KNN

Reference :<https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>

2.11.4 How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

Step 1: Select the number K of the neighbors

Step 2: Calculate the Euclidean distance of K number of neighbors

Step 3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step 4: Among these K neighbors, count the number of the data points in each category.

Step 5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step 6: Our model is ready. Suppose we have a new data point and we need to put it in the required category. Consider the image given below:

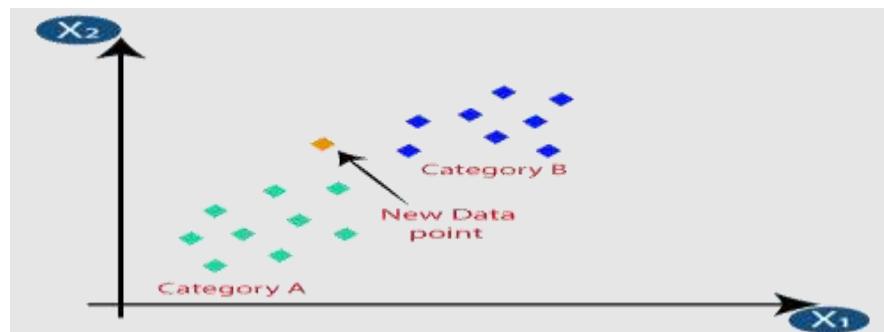


Fig: New data point

Reference : <https://static.javatpoint.com/tutorial/machine-learning/images/k-nearest-neighbor-algorithm-for-machine-learning3.png>

- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:

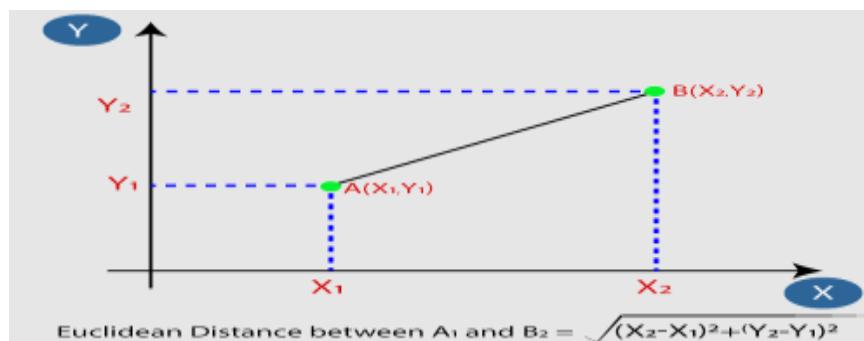


Fig: Euclidean distance calculation

Reference: <https://static.javatpoint.com/tutorial/machine-learning/images/k-nearest-neighbor-algorithm-for-machine-learning4.png>

- By calculating the Euclidean distance we get the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the image below:

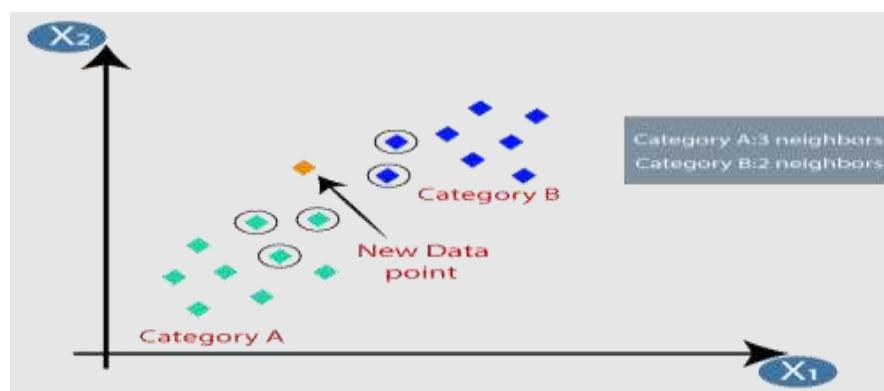


Fig: Finding of Nearest neighbors

Reference: <https://static.javatpoint.com/tutorial/machine-learning/images/k-nearest-neighbor-algorithm-for-machine-learning4.png>

- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm

- It is simple to implement.
- It is robust to the noisy training data.
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

2.12 Practical: K-NN using Python – Scikit Learn library

We will use the same dataset which was used earlier of iris flower and apply K NN Algorithm. Also we will see how to plot a decision boundary with respect to 3 categories using ListedColorMap class from matplotlib colors library. Here we can fine tune the parameters accordingly

The link for project is given below:

https://github.com/Code-Unnati/Module-III/blob/master/Machine%20Learning%20Models/Decision_Boundary_for_KNN.ipynb

2.13 Decision Trees

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

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

The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question and based on the answer (Yes/No), it further splits the tree into subtrees.

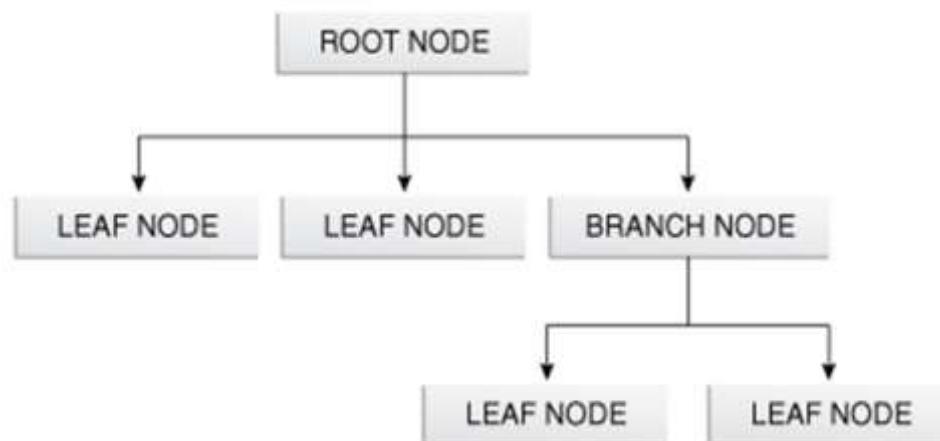


Fig: [General structure of a decision tree](#)

2.13.1 Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.

The logic behind the decision tree can be easily understood because it shows a tree-like structure.

2.13.2 Decision Tree Terminologies

Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Branch/Sub Tree: A tree formed by splitting the tree.

Pruning: Pruning is the process of removing the unwanted branches from the tree.

Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

2.13.3 How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

Step 1: Begin the tree with the root node, says S, which contains the complete dataset.

Step 2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step 3: Divide the S into subsets that contains possible values for the best attributes.

Step 4: Generate the decision tree node, which contains the best attribute.

Step 5: Recursively make new decision trees using the subsets of the dataset created in step 3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:



Fig: [Example](#)

Reference: <https://static.javatpoint.com/tutorial/machine-learning/images/decision-tree-classification-algorithm2.png>

We understood that how to construct a decision tree but the main thing is that how we can select the best node for splitting, right! So, let us explore it.

2.13.4 Gini Impurity or Entropy

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM. By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

1. Information Gain
2. Gini Index

Gini index and entropy are the criteria for calculating information gain. Decision tree algorithms use information gain to split a node.

Both gini and entropy are measures of impurity of a node. A node having multiple classes is impure whereas a node having only one class is pure.

Entropy in statistics is analogous to entropy in thermodynamics where it signifies disorder. If there are multiple classes in a node, there is disorder in that node.

$$Gini = 1 - \sum_{i=1}^n p^2(c_i)$$

$$Entropy = \sum_{i=1}^n -p(c_i) \log_2(p(c_i))$$

where $p(c_i)$ is the probability/percentage of class c_i in a node.

Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm. An attribute with the low Gini index should be preferred as compared to the high Gini index. It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits. Information gain is the entropy of parent node minus sum of weighted entropies of child nodes. Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute. It calculates how much information a feature provides us about a class. According to the value of information gain, we split the node and build the decision tree. A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

$$\text{Information Gain} = \text{Entropy}(S) - [(\text{Weighted Avg}) * \text{Entropy (each feature)}]$$

Weight of a child node is number of samples in the node/total samples of all child nodes. Similarly, information gain is calculated with gini score.

2.14 Practical: Decision Tree Classifier using Sklearn

The iris dataset is a classic and very easy multi-class classification dataset. The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Let us apply Decision Tree Classifier on the same.

The link for the project is given below:

https://github.com/Code-Unnati/Module-III/blob/master/Machine%20Learning%20Models/Decision_Tree.ipynb

2.15 Random Forest: Concept

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like scikit-learn).

Features of a Random Forest Algorithm

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of overfitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

Applying decision trees in random forest

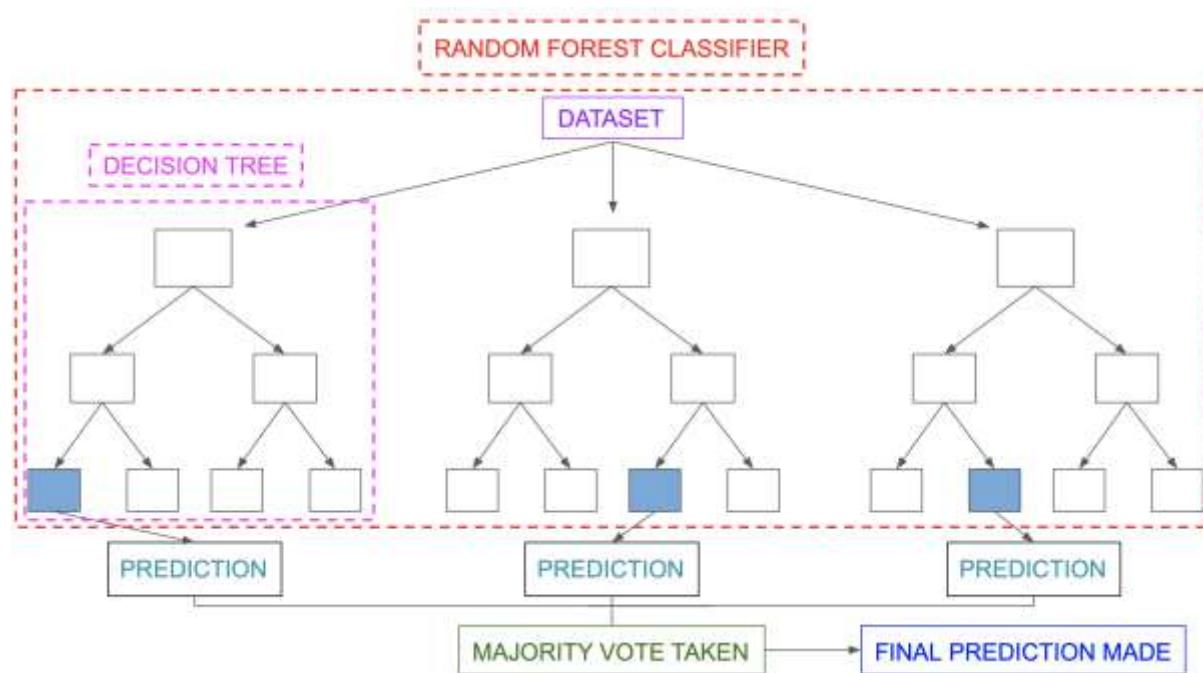
The main difference between the decision tree algorithm and the random forest algorithm is that establishing root nodes and segregating nodes is done randomly in the latter. The random forest employs the bagging method to generate the required prediction.

Bagging involves using different samples of data (training data) rather than just one sample. A training dataset comprises observations and features that are used for making predictions. The decision trees produce different outputs, depending on the training data fed to the random forest algorithm. These outputs will be ranked, and the highest will be selected as the final output.

2.15.1 Classification in random forests

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes.

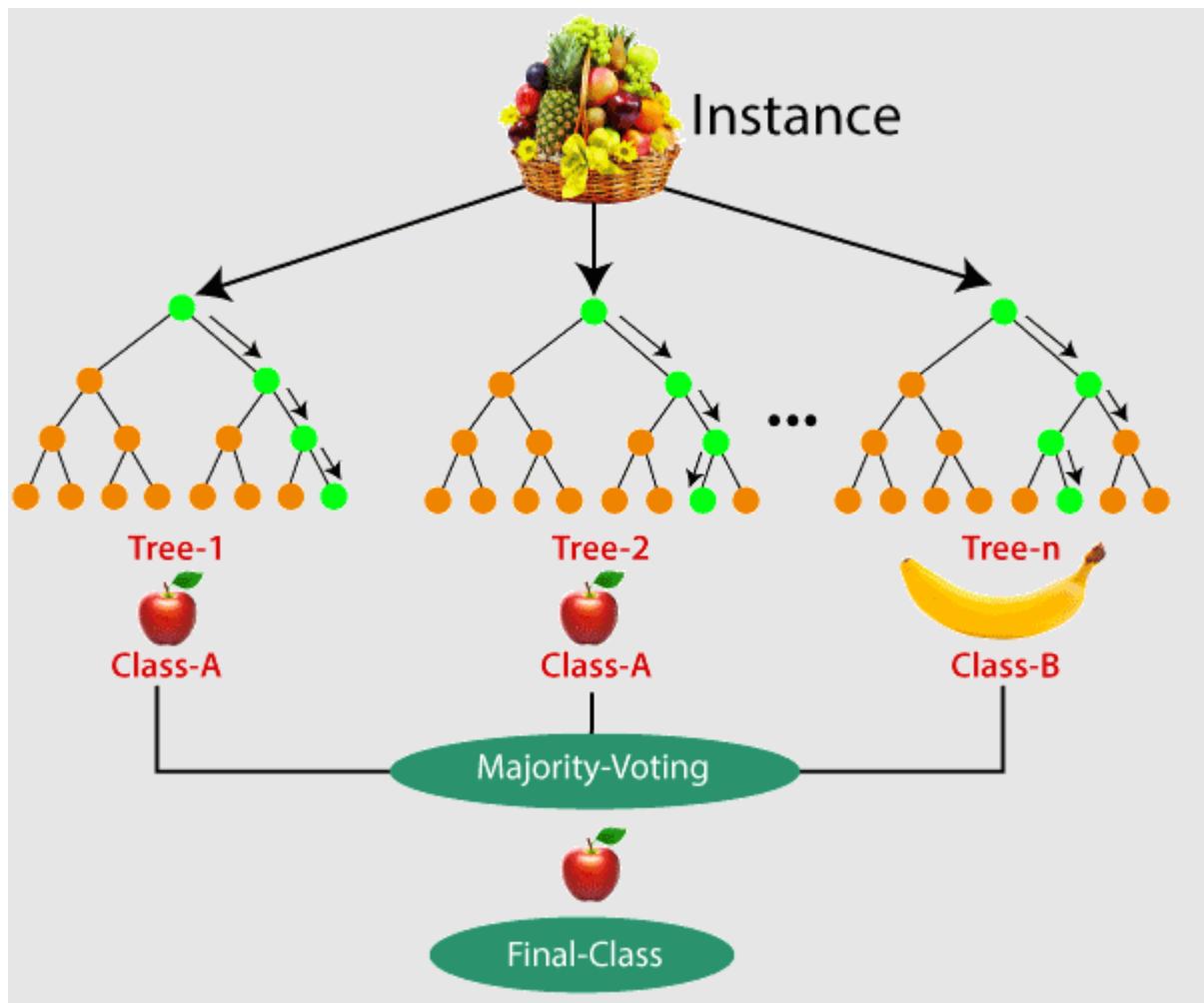
A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system. The diagram below shows a simple random forest classifier.



Reference: https://miro.medium.com/v2/resize:fit:5752/1*5dq_1hnqkboZTcKFfwbO9A.png

Let's take an example of a training dataset consisting of various fruits such as bananas, apples, pineapples, and mangoes. The random forest classifier divides this dataset into subsets. These subsets are given to every decision tree in the random forest system. Each decision tree produces its specific output. For example, the prediction for trees 1 and 2 is *apple*.

Another decision tree (n) has predicted *banana* as the outcome. The random forest classifier collects the majority voting to provide the final prediction. The majority of the decision trees have chosen *apple* as their prediction. This makes the classifier choose *apple* as the final prediction.



Reference: <https://static.javatpoint.com/tutorial/machine-learning/images/random-forest-algorithm2.png>

2.15.2 How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step 1: Select random K data points from the training set.

Step 2: Build the decision trees associated with the selected data points (Subsets).

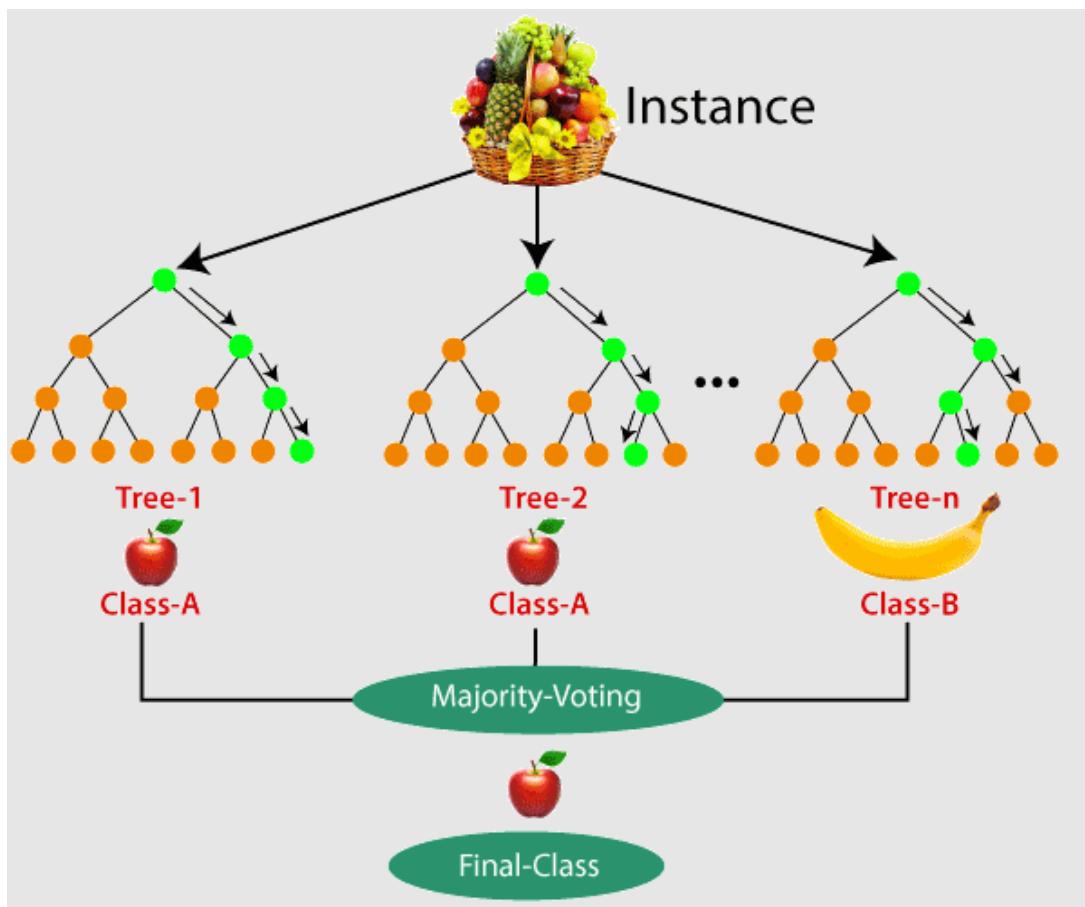
Step 3: Choose the number N for decision trees that you want to build.

Step 4: Repeat Step 1 & 2.

Step 5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:



Reference: <https://static.javatpoint.com/tutorial/machine-learning/images/random-forest-algorithm2.png>

2.16 Bagging & Boosting in Machine Learning

The basic idea is to learn a set of classifiers (experts) and to allow them to vote. **Bagging** and **Boosting** are two types of **Ensemble Learning**. These two decrease the variance of a single estimate as they combine several estimates from different models. So the result may be a model with higher stability. Let's understand these two terms in a glimpse.

- **Bagging:** It is a homogeneous weak learners' model that learns from each other independently in parallel and combines them for determining the model average.
- **Boosting:** It is also a homogeneous weak learners' model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

Implementation Steps of Bagging

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

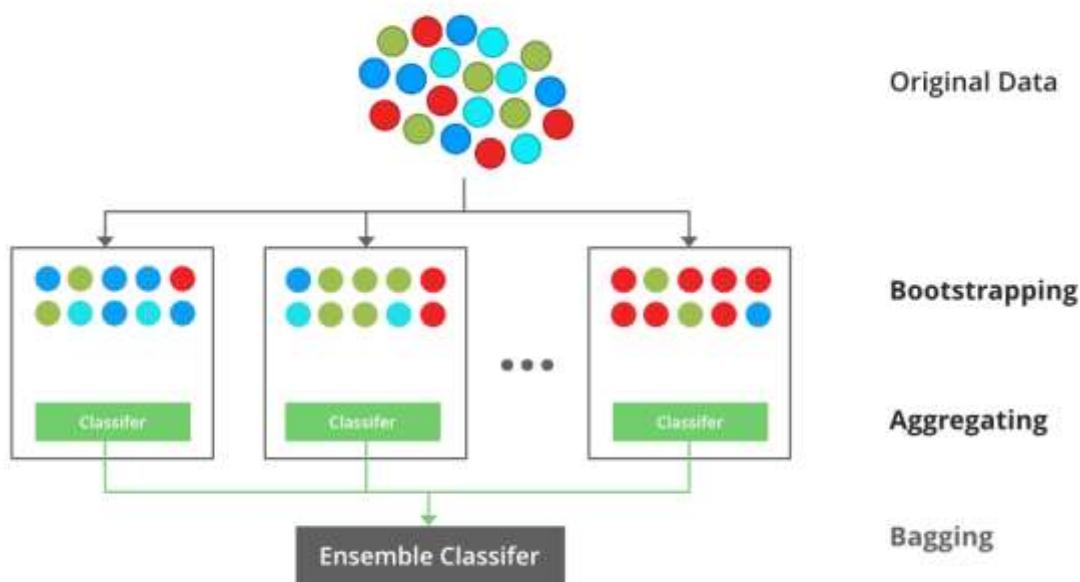


Fig:[Bootstrap aggregating \(Bagging\)](#)

Reference: <https://media.geeksforgeeks.org/wp-content/uploads/20210707140912/Bagging.png>

Example of Bagging

The Random Forest model uses Bagging, where decision tree models with higher variance are present. It makes random feature selection to grow trees. Several random trees make a Random Forest.

Boosting

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

Boosting Algorithms

There are several boosting algorithms. The original ones, proposed by **Robert Schapire** and **Yoav Freund** were not adaptive and could not take full advantage of the weak learners. Schapire and Freund then developed AdaBoost, an adaptive boosting algorithm that won the prestigious Gödel Prize. AdaBoost was the first really successful boosting algorithm developed for the purpose of binary classification. AdaBoost is short for Adaptive Boosting and is a very popular boosting technique that combines multiple “weak classifiers” into a single “strong classifier”.

Algorithm:

- *Initialise the dataset and assign equal weight to each of the data point.*
- *Provide this as input to the model and identify the wrongly classified data points.*
- *Increase the weight of the wrongly classified data points and decrease the weights of correctly classified data points. And then normalize the weights of all data points.*
- *if (got required results)*
 - Goto step 5*
 - else*
 - Goto step 2*
 - End*

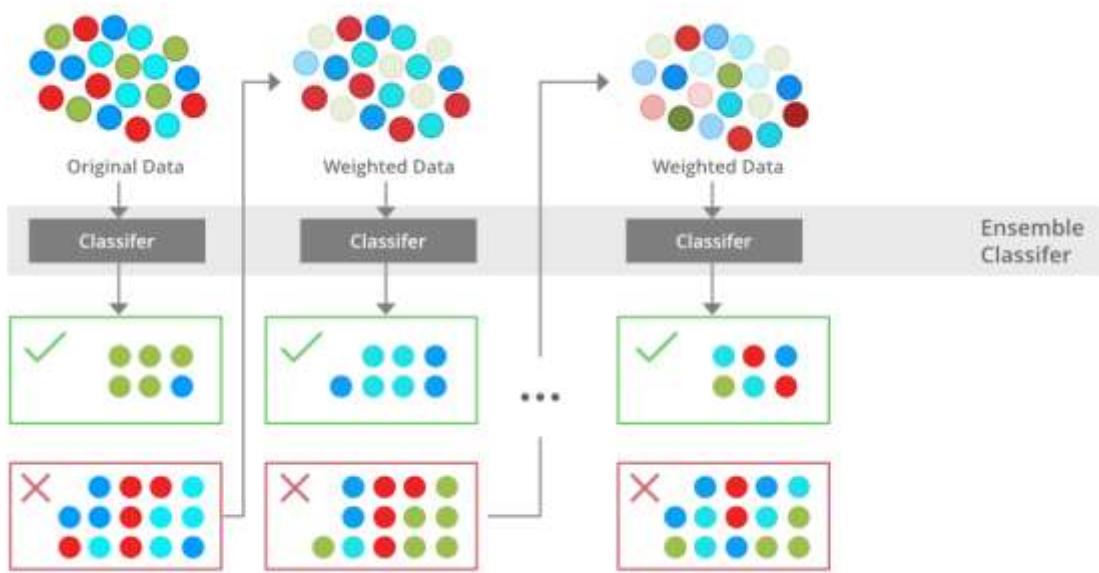


Fig:Boosting Weak Learners

Reference: <https://media.geeksforgeeks.org/wp-content/uploads/20210707140911/Boosting.png>

2.17 Practical: Random Forest using Python- Scikit Learn library

The link for project is given below:

2.17.1. Simple to Complex Forest: Bias Variance Tradeoff

The link for Project is given below:

2.18 Support Vector Machine: Concept

Support Vector Machines (SVMs in short) are machine learning algorithms that are used for classification and regression purposes. SVMs are one of the powerful machine learning algorithms for classification, regression and outlier detection purposes. An SVM classifier builds a model that assigns new data points to one of the given categories. Thus, it can be viewed as a non-probabilistic binary linear classifier.

SVMs can be used for linear classification purposes. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using the kernel trick. It enables us to implicitly map the inputs into high dimensional feature spaces.

2.18.1 Linear SVM Classification

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

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

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

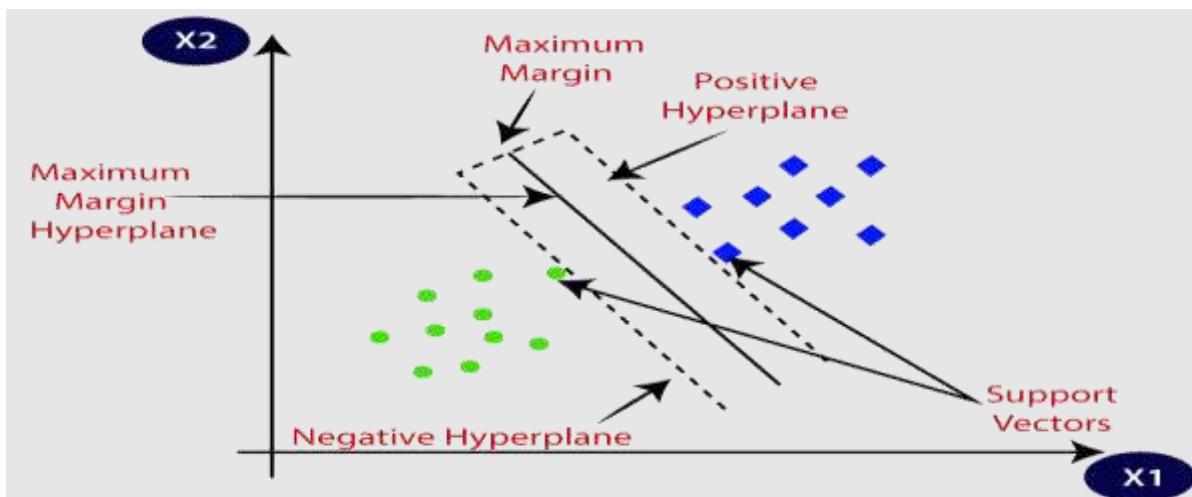


Fig: Support Vector Machine

Reference: <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>

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

vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:

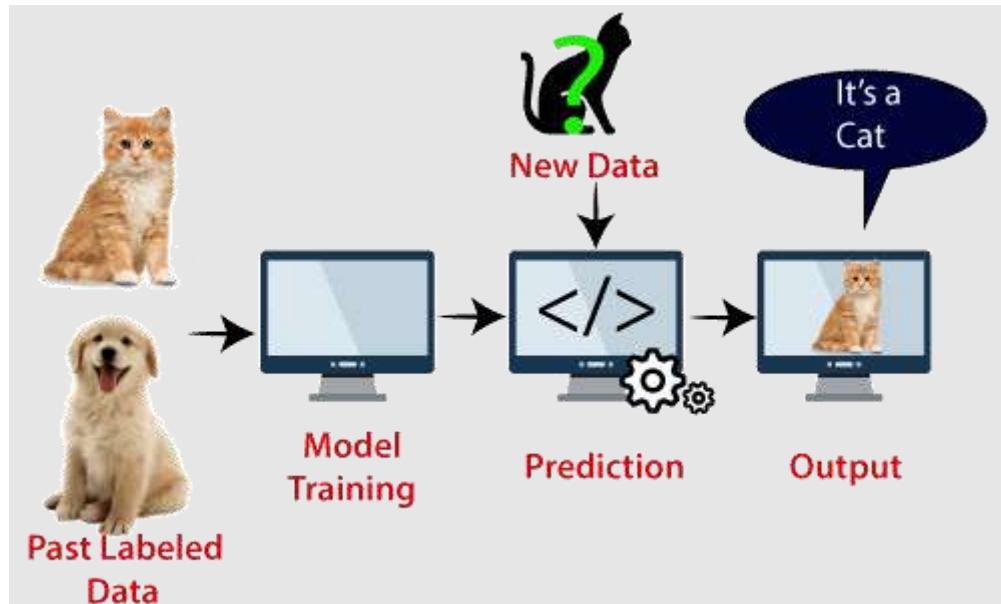


Fig: Support Vector Machine Example

Reference: <https://cdn.inblog.in/user/uploads/x7PHukCnjBmaGEAQTDUKbiwszsZvJt.png>

SVM algorithm can be used for Face detection, image classification, text categorization, etc.

2.18.2 Types of SVM

SVM can be of two types:

- **Linear SVM**

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier

- **Non-linear SVM**

Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

2.18.3 How does SVM works?

Linear SVM:

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x_1 and x_2 . We want a classifier that can classify the pair (x_1, x_2) of coordinates in either green or blue. Consider the below image:

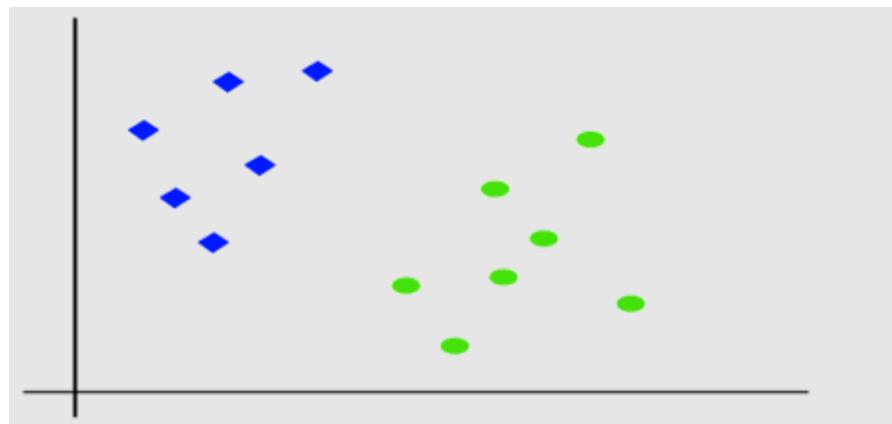


Fig: Dataset with Green Blue tags

Reference: <https://static.javatpoint.com/tutorial/machine-learning/images/support-vector-machine-algorithm3.png>

So, as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

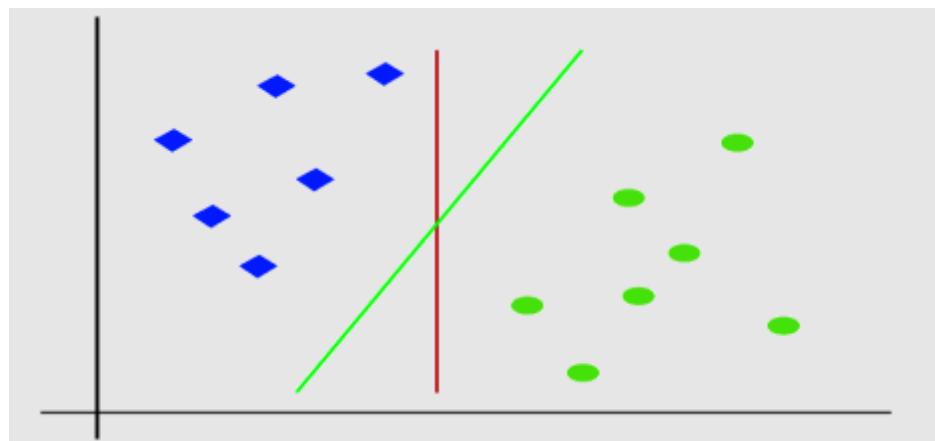


Fig: Multiple lines to separate Classes

Reference: <https://static.javatpoint.com/tutorial/machine-learning/images/support-vector-machine-algorithm4.png>

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is

to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

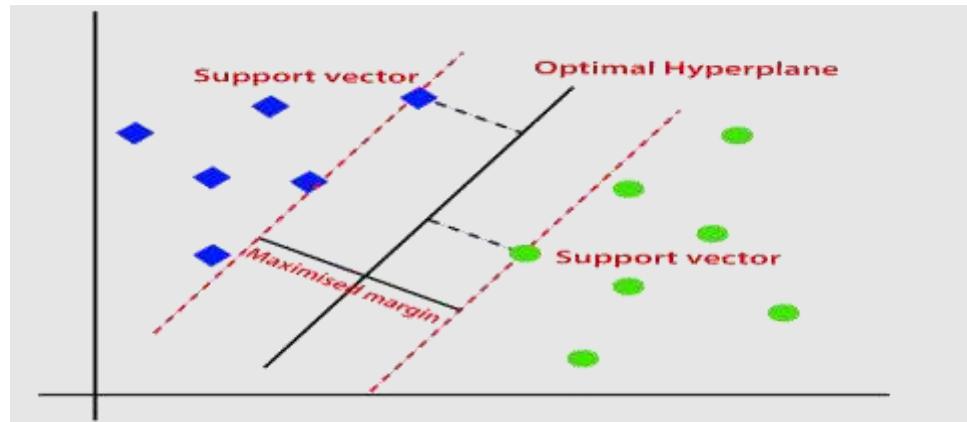


Fig: Hyperplane

Reference: <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>

Support Vectors and Margins

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

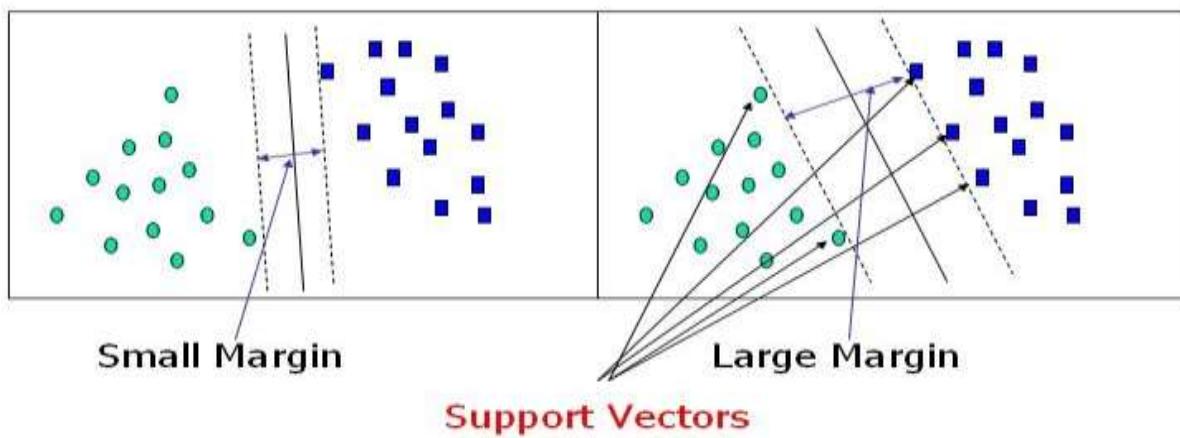


Fig: Support Vectors

Reference: https://miro.medium.com/max/700/0*ecA4Ls8kBYSM5nza.jpg

Terminologies used in SVM

The points closest to the hyperplane are called as the support vector points and the distance of the vectors from the hyperplane are called the margins.

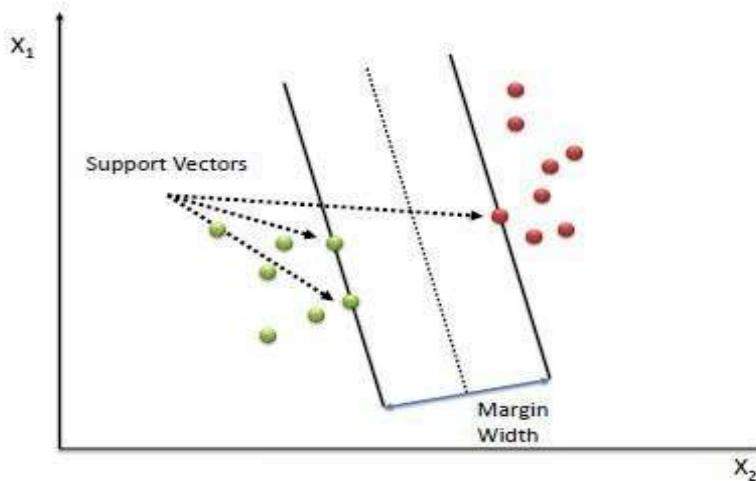


Fig: Support vectors and margins

Reference: <https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589>

The basic intuition to develop over here is that more the farther SV points, from the hyperplane, more is the probability of correctly classifying the points in their respective region or classes. SV points are very critical in determining the hyperplane because if the position of the vectors changes the hyperplane's position is altered. Technically this hyperplane can also be called as margin maximizing hyperplane.

Hard margin SVM

Assume 3 hyperplanes namely (π , π^+ , π^-) such that ' π^+ ' is parallel to ' π ' passing through the support vectors on the positive side and ' π^- ' is parallel to ' π ' passing through the support vectors on the negative side.

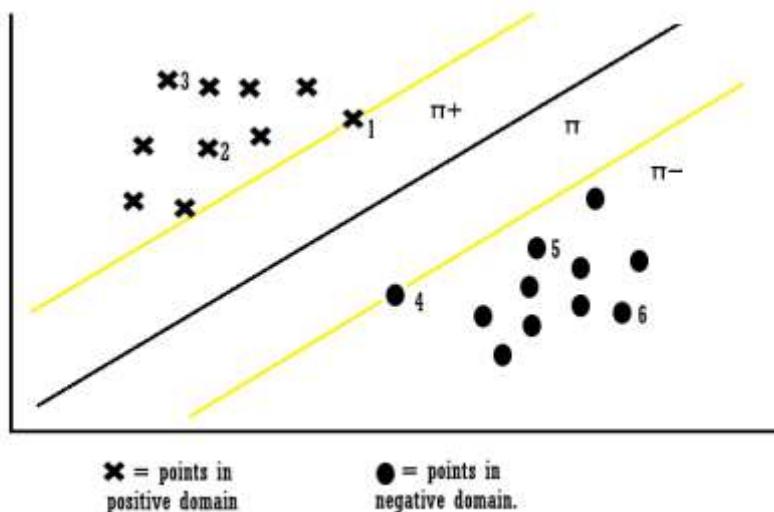


Fig: Support vectors and Hyperplanes

Reference: https://miro.medium.com/max/687/1*ppsJ51l8o5kTC1q2opSjwQ.png

The equations of each hyperplane can be considered as:

$$\begin{aligned}\pi &= b + w^T X = 0 \\ \pi^+ &= b + w^T X = 1 \\ \pi^- &= b + w^T X = -1\end{aligned}$$

for the point X1:

$$\begin{aligned}y_1 &= 1 \\ y_1(w^T x_1 + b) &= 1\end{aligned}$$

Explanation: when the point X1 we can say that point lies on the hyperplane and the equation determines that the product of our actual output and the hyperplane equation is 1 which means the point is correctly classified in the positive domain.

for the point X3:

$$\begin{aligned}y_1 &= 1 \\ y_1(w^T x_1 + b) &> 1\end{aligned}$$

Explanation: when the point X3 we can say that point lies away from the hyperplane and the equation determines that the product of our actual output and the hyperplane equation is greater 1 which means the point is correctly classified in the positive domain.

for the point X4:

$$\begin{aligned}y_1 &= -1 \\ y_1(w^T x_1 + b) &= 1\end{aligned}$$

Explanation: when the point X4 we can say that point lies on the hyperplane in the negative region and the equation determines that the product of our actual output and the hyperplane equation is equal to 1 which means the point is correctly classified in the negative domain.

for the point X6:

$$\begin{aligned}y_1 &= -1 \\ y_1(w^T x_1 + b) &> 1\end{aligned}$$

Explanation: when the point X6 we can say that point lies away from the hyperplane in the negative region and the equation determines that the product of our actual output and the hyperplane equation is greater 1 which means the point is correctly classified in the negative domain.

Let's look into the constraints which are not classified:

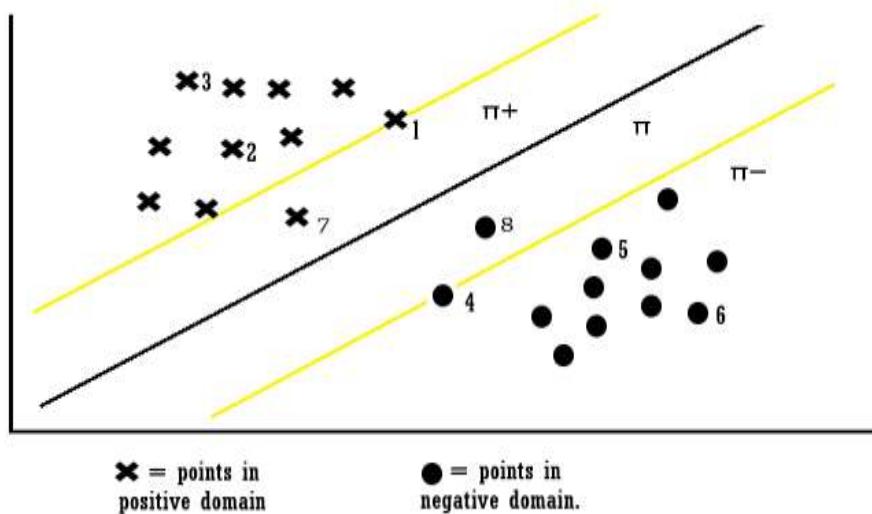


Fig: Unclassified Constraints
Reference: https://miro.medium.com/max/687/1*ppsJ51l8o5kTC1q2opSiwQ.png

for point X_7 :

$$\begin{aligned} y_1 &= 1 \\ y_1(w^T x_1 + b) & \\ (1)(1 <) \end{aligned}$$

Explanation: When $X_i = 7$ the point is classified incorrectly because for point 7 the $w^T + b$ will be smaller than one and this violates the constraints. So we found the misclassification because of constraint violation. Similarly, we can also say for points $X_i = 8$.

Thus, from the above examples, we can conclude that for any point X_i ,

if $y_i(w^T x_i + b) \geq 1$:

then X_i is correctly classified

else:

X_i is incorrectly classified.

So, we can see that if the points are linearly separable then only our hyperplane is able to distinguish between them and if any outlier is introduced then it is not able to separate them. So, these type of SVM is called as **hard margin SVM** (since we have very strict constraints to correctly classify each and every datapoint).

Soft margin SVM

We basically consider that the data is linearly separable and this might not be the case in real life scenario. We need an update so that our function may skip few outliers and be able to classify almost linearly separable points. For this reason, we introduce a new Slack variable (ξ) which is called X_i .

if we introduce ξ it into our previous equation we can rewrite it as

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$

if $\xi_i = 0$,

the points can be considered as correctly classified.

else:

$\xi_i > 0$, Incorrectly classified points.

so, if $\xi_i > 0$ it means that X_i (variables) lies in incorrect dimension, thus we can think of ξ_i as an error term associated with X_i (variable). The average error can be given as;

$$\frac{1}{n} \sum_{i=1}^n \xi_i$$

thus, our objective, mathematically can be described as;

$$\underset{w,b}{\text{minimize}} \frac{1}{2} \|w\|^2 + \sum_{i=1}^n \xi_i$$

$$\text{such that } y_i(w^T X_i + b) \geq 1 - \xi_i \quad \text{For all } i = 1, 2, \dots, n$$

where $\xi_i = \zeta_i$

2.19 Practical: Support vector Machine Using Python

The link for project is given below:

2.20 Ensemble learning

The literary meaning of the word Ensemble is a group. Ensemble methods involve group of predictive models to achieve better accuracy and model stability. Ensemble methods are known to impart supreme boost to tree based models.

Like every other model, a tree based algorithm also suffers from the plague of ****bias**** and ****variance****.

Decision trees are prone to overfitting. Normally, as you increase the complexity of your model, in this case, the decision tree, you will see a reduction in training error due to lower bias in the model. As you continue to make your model more complex, you end up over-fitting your model and your model will start suffering from high variance.

A champion model should maintain a balance between these two types of errors. This is known as the trade-off management of bias-variance errors.

Ensemble learning is one way to tackle bias-variance trade-off.

There are various ways to ensemble *weak* learners to come up with *strong* learners:

1. Bagging
2. Boosting
3. Stacking

1. Bagging

Bagging is an ensemble technique used to reduce the variance of our predictions by combining the result of multiple classifiers modeled on different sub-samples of the same data set. The following figure will make it clearer:

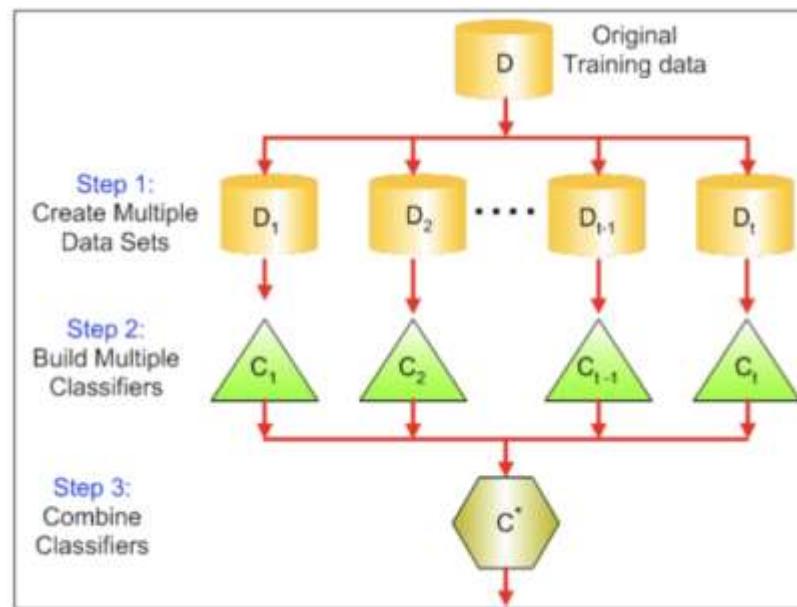


Fig:Bagging

Reference: https://en.wikipedia.org/wiki/Bootstrap_aggregating

The steps followed in bagging are:

1. **Create Multiple Datasets:** Sampling is done with replacement on the original data and new datasets are formed.
2. **Build Multiple Classifiers:** Classifiers are built on each data set. Generally the same classifier is modeled on each data set and predictions are made.
3. **Combine Classifiers:** The predictions of all the classifiers are combined using a mean, median or mode value depending on the problem at hand.

The combined values are generally more robust than a single model.

Note that, here the number of models built is not a hyper-parameters. Higher number of models are always better or may give similar performance than lower numbers.

Important: It can be theoretically shown that the variance of the combined predictions are reduced to $1/n$ (n : number of classifiers) of the original variance, under some assumptions. (Think Central Limit Theorem)

There are various implementations of bagging models. Random forest is one of them and we'll discuss it next.

Random Forest

In Random Forest, we grow multiple trees as opposed to a single tree in CART model.

- We construct trees from the subsets of the original dataset. These subsets can have a fraction of the columns as well as rows.
- To classify a new object based on attributes, each tree gives a classification and we say that the tree “votes” for that class.
- The forest chooses the classification having the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.

Random Forest Simplified

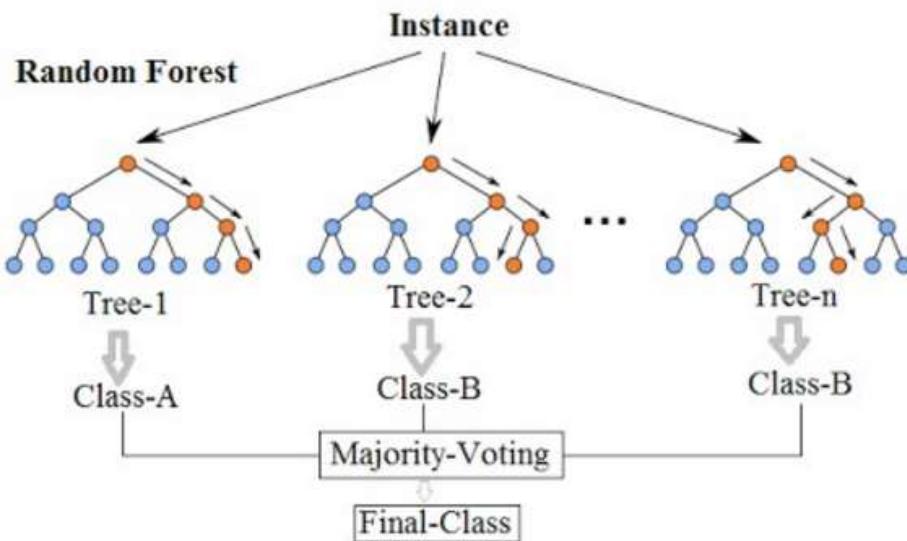


Fig:[Random Forest](#)

Reference: <https://www.tibco.com/reference-center/what-is-a-random-forest>

It works in the following manner:

- Assume number of rows in the training set is N. Then, a sample of $n < N$ rows is taken at random but **with replacement.** This sample will be the training set for growing the tree.
- If there are M input variables, a number $m < M$ is specified such that at each node, m variables are selected at random out of the M. The best split on these m is used to split the node. The value of m is held constant while we grow the forest.
- Each tree is grown to the largest extent possible and there is no pruning.
- Predict new data by aggregating the predictions of the n tree trees (i.e., majority votes for classification, average for regression).

Advantages

- This algorithm can solve both type of problems i.e. classification and regression and does a decent estimation at both fronts.
- RF has the power of handling large datasets with higher dimensionality. It can handle thousands of input variables and identify most significant variables so it is considered as one of the dimensionality reduction methods. Further, the model outputs Importance of variable, which can be a very handy feature (on some random data set).

- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data is missing.
- It has methods for balancing errors in data sets where classes are imbalanced.
- The capabilities of the above can be extended to unlabeled data, leading to unsupervised clustering, data views and outlier detection.
- Random Forest involves sampling of the input data with replacement called as bootstrap sampling. Here one third (say) of the data is not used for training and can be used to testing. These are called the out of bag samples. Error estimated on these out of bag samples is known as out of bag error. Out-of-bag estimate is as accurate as using a test set of the same size as the training set. Therefore, using the out-of-bag error estimate removes the need for a set aside test set.

Disadvantages

- It surely does a good job at classification but not as good as for regression problem as it does not give precise continuous nature predictions. In case of regression, it doesn't predict beyond the range in the training data, and that they may over-fit data sets that are particularly noisy.
- Random Forest can feel like a black box approach for statistical modelers – you have very little control on what the model does. You can at best – try different parameters and random seeds!

2. Boosting

Boosting fit a sequence of weak learners – models that are only slightly better than random guessing, such as small decision trees – to weighted versions of the data. More weight is given to examples that were misclassified by earlier rounds.

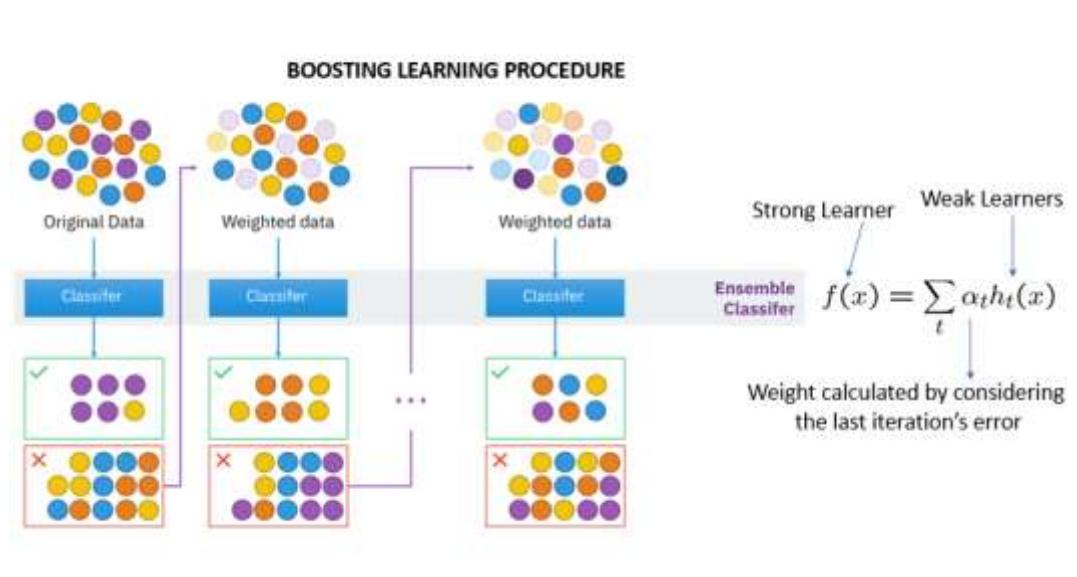


Fig:Boosting

Reference: <https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/>

There are many boosting algorithms which impart additional boost to model's accuracy:

1. Gradient Boosting Machine
2. XGBoost
3. AdaBoost
4. LightGBM
5. CatBoost

Gradient Boosting Machine

Gradient Boosting Machine (GBM) builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

How does it work?

- Let's take a dataset $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)\}$
- Choose a loss function, let's say MSE.
- Fit a naive model on the dataset, a simple tree or just take y^- , call this model $F_0(x)$

First iteration

- Get residuals of all predictions, $r_{i1}(x) = y_i - F_0(x_i)$
- Fit a model (can be regression tree) on residuals $\{(x_1, r_{11}), (x_2, r_{21}), (x_3, r_{31}), \dots, (x_n, r_{n1})\}$, call this model $h_1(x)$
- New predictor is $F_1(x) = F_0(x) + \gamma_1 h_1(x)$. Find γ_1 which minimizes MSE.
- Second iteration
- Get residuals of all predictions, $r_{i2}(x) = y_i - F_1(x)$
- Fit a model (can be regression tree) on residuals $\{(x_1, r_{12}), (x_2, r_{22}), (x_3, r_{32}), \dots, (x_n, r_{n2})\}$, call this model $h_2(x)$
- New predictor is $F_2(x) = F_1(x) + \gamma_2 h_2(x)$. Find γ_2 which minimizes MSE.
- and so on...
- Get residuals of all predictions, $r_{im}(x) = y_i - F_{m-1}(x)$
- Fit a model (can be regression tree) on residuals $\{(x_1, r_{1m}), (x_2, r_{2m}), (x_3, r_{3m}), \dots, (x_n, r_{nm})\}$, call this model $h_m(x)$
- Final predictor is $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$. Find γ_m which minimizes MSE.

XGBoost

Extreme Gradient Boosting (XGBoost) is just an extension of gradient boosting with the following added advantages:

- Regularization: Standard GBM implementation has no regularization like XGBoost, therefore it also helps to reduce overfitting. In fact, XGBoost is also known as ‘regularized boosting’ technique.
- Parallel Processing: XGBoost implements parallel processing and is blazingly faster as compared to GBM. But hang on, we know that boosting is sequential process so how can it be parallelized? We know that each tree can be built only after the previous one, but to make a tree it uses all the cores of the system. XGBoost also supports implementation on Hadoop.
- High Flexibility: XGBoost allow users to define custom optimization objectives and evaluation criteria. This adds a whole new dimension to the model and there is no limit to what we can do.
- Handling Missing Values: XGBoost has an in-built routine to handle missing values. User is required to supply a different value than other observations and pass that as a parameter. XGBoost tries different things as it encounters a missing value on each node and learns which path to take for missing values in future.
- Tree Pruning: A GBM would stop splitting a node when it encounters a negative loss in the split. Thus it is more of a greedy algorithm. XGBoost on the other hand make splits upto the max_depth specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain.
- Another advantage is that sometimes a split of negative loss say -2 may be followed by a split of positive loss +10. GBM would stop as it encounters -2. But XGBoost will go deeper and it will see a combined effect of +8 of the split and keep both.
- Built-in Cross-Validation: XGBoost allows user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run. This is unlike GBM where we have to run a grid-search and only a limited values can be tested.
- Continue on Existing Model: User can start training an XGBoost model from its last iteration of previous run. This can be of significant advantage in certain specific applications. GBM implementation of sklearn also has this feature so they are even on this point.

3. Stacking

Stacking or Stacked Generalization is an ensemble technique. It uses a meta-learning algorithm to learn how to best combine the predictions from two or more base machine learning algorithms. The benefit of stacking is that it can harness the capabilities of a range of well-performing models on a classification or regression task and make predictions that have better performance than any single model in the ensemble.

Given multiple machine learning models that are skillful on a problem, but in different ways, how do you choose which model to use (trust)?

The approach to this question is to use another machine learning model that learns when to use or trust each model in the ensemble.

Unlike bagging, in stacking, the models are typically different (e.g. not all decision trees) and fit on the same dataset (e.g. instead of samples of the training dataset)

Unlike boosting, in stacking, a single model is used to learn how to best combine the predictions from the contributing models (e.g. instead of a sequence of models that correct the predictions of prior models).

The architecture of a stacking model involves two or more basemodels, often referred to as level-0 models, and a meta-model that combines the predictions of the base models, referred to as a level-1 model.

Level-0 Models (Base-Models): Models fit on the training data and whose predictions are compiled.

Level-1 Model (Meta-Model): Model that learns how to best combine the predictions of the base models.

The meta-model is trained on the predictions made by base models on out-of-sample data. That is, data not used to train the base models is fed to the base models, predictions are made, and these predictions, along with the expected outputs, provide the input and output pairs of the training dataset used to fit the meta-model.

The outputs from the base models used as input to the meta-model may be real value in the case of regression, and probability values, probability like values, or class labels in the case of classification.

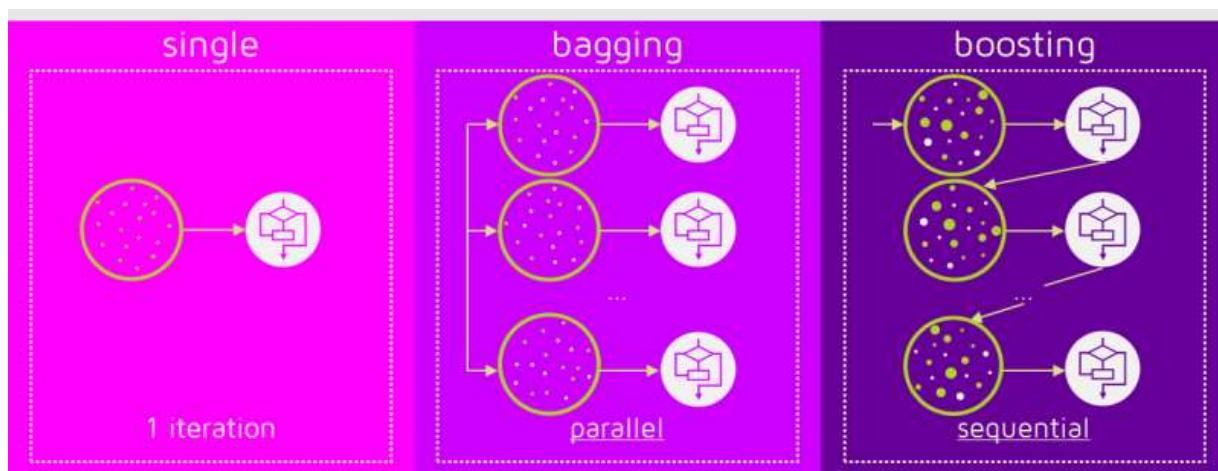


Fig:Bagging-boosting-and-stacking

Reference: <https://www.analyticsvidhya.com/blog/2023/01/ensemble-learning-methods-bagging-boosting-and-stacking/>

2.21 Practical: Ensemble Learning Using Python

The link for project is given below:

Unit 3: Unsupervised Machine Learning

Learning Outcomes:

- Understand the basics of Clustering and types of Clustering
- Understand the basic of K Means Clustering.
- Able to Perform Clustering using different algorithms.
- Able to perform Dimensionality Reduction using PCA
- Able to use LDA as Classifier

3.1 Unsupervised Learning

Unsupervised transformations of a dataset are algorithms that create a new representation of the data which might be easier for humans or other machine learning algorithms to understand compared to the original representation of the data. A common application of unsupervised transformations is dimensionality reduction, which takes a high-dimensional representation of the data, consisting of many features, and finds a new way to represent this data that summarizes the essential characteristics with fewer features. A common application for dimensionality reduction is reduction to two dimensions for visualization purposes.

Another application for unsupervised transformations is finding the parts or components that “make up” the data. An example of this is topic extraction on collections of text documents. Here, the task is to find the unknown topics that are talked about in each document, and to learn what topics appear in each document. This can be useful for tracking the discussion of themes like elections, gun control, or pop stars on social media.

Clustering algorithms, on the other hand, partition data into distinct groups of similar items. Consider the example of uploading photos to a social media site. To allow you to organize your pictures, the site might want to group together pictures that show the same person. However, the site doesn’t know which pictures show whom, and it doesn’t know how many different people appear in your photo collection. A sensible approach would be to extract all the faces and divide them into groups of faces that look similar. Hopefully, these correspond to the same person, and the images can be grouped together for you.

3.1.1 Challenges in Unsupervised Learning

A major challenge in unsupervised learning is evaluating whether the algorithm learned something useful. Unsupervised learning algorithms are usually applied to data that does not contain any label information, so we don't know what the right output should be. Therefore, it is very hard to say whether a model "did well." For example, our hypothetical clustering algorithm could have grouped together all the pictures that show faces in profile and all the full-face pictures. This would certainly be a possible way to divide a collection of pictures of people's faces, but it's not the one we were looking for. However, there is no way for us to "tell" the algorithm what we are looking for, and often the only way to evaluate the result of an unsupervised algorithm is to inspect it manually. As a consequence, unsupervised algorithms are used often in an exploratory setting, when a data scientist wants to understand the data better, rather than as part of a larger automatic system. Another common application for unsupervised algorithms is as a pre-processing step for supervised algorithms. Learning a new representation of the data can sometimes improve the accuracy of supervised algorithms, or can lead to reduced memory and time consumption.

3.1.2 Why Unsupervised Learning?

Here, are prime reasons for using Unsupervised Learning:

- Unsupervised machine learning finds all kind of unknown patterns in data.
- Unsupervised methods help you to find features which can be useful for categorization.
- It takes place in real time, so all the input data is to be analysed and labelled in the presence of learners.
- It is easier to get unlabelled data from a computer than labelled data, which needs manual intervention.

3.1.3 Unsupervised Learning can be classified into two categories:

Parametric Unsupervised Learning

In this case, we assume a parametric distribution of data. It assumes that sample data comes from a population that follows a probability distribution based on a fixed set of parameters. Theoretically, in a normal family of distributions, all members have the same shape and are *parameterized* by mean and standard deviation. That means if you know the mean and standard deviation, and that the distribution is normal, you know the probability of any future observation. Parametric Unsupervised Learning involves construction of Gaussian Mixture Models and using Expectation-Maximization algorithm to predict the class of the sample in question. This case is much harder than the

standard supervised learning because there are no answer labels available and hence there is no correct measure of accuracy available to check the result.

Non-parametric Unsupervised Learning

In non-parameterized version of unsupervised learning, the data is grouped into clusters, where each cluster (hopefully) says something about categories and classes present in the data. This method is commonly used to model and analyze data with small sample sizes. Unlike parametric models, nonparametric models do not require the modeler to make any assumptions about the distribution of the population, and so are sometimes referred to as a distribution-free method.

Let us get started with our first type of unsupervised machine learning algorithm called clustering.

3.2 Introduction to Clustering

Imagine that you are a Data Scientist working for a retail company and your boss requests for the customers' segmentation into the following groups: low, average, medium, or platinum customers based on spending behaviour for targeted marketing purposes and product recommendations.

Knowing that there is no such historical label associated with those customers, how is it possible to categorize them? This is where clustering can help.

3.2.1 Clustering

It is an unsupervised machine-learning technique used to group unlabeled data into similar categories or clusters.

OR

In Simple words, Clustering is a type of unsupervised learning wherein data points are grouped into different sets based on their degree of similarity.

Clustering can be used in many areas, including machine learning, computer graphics, pattern recognition, image analysis, information retrieval, bioinformatics, and data compression. A broad range of industries use clustering, from airlines to healthcare and beyond.

Clustering itself is not one specific algorithm but the general task to be solved. You can achieve this goal using various algorithms that differ significantly in their understanding of what constitutes a cluster and how to find them efficiently.

For example— There are 3 clusters in the below picture.

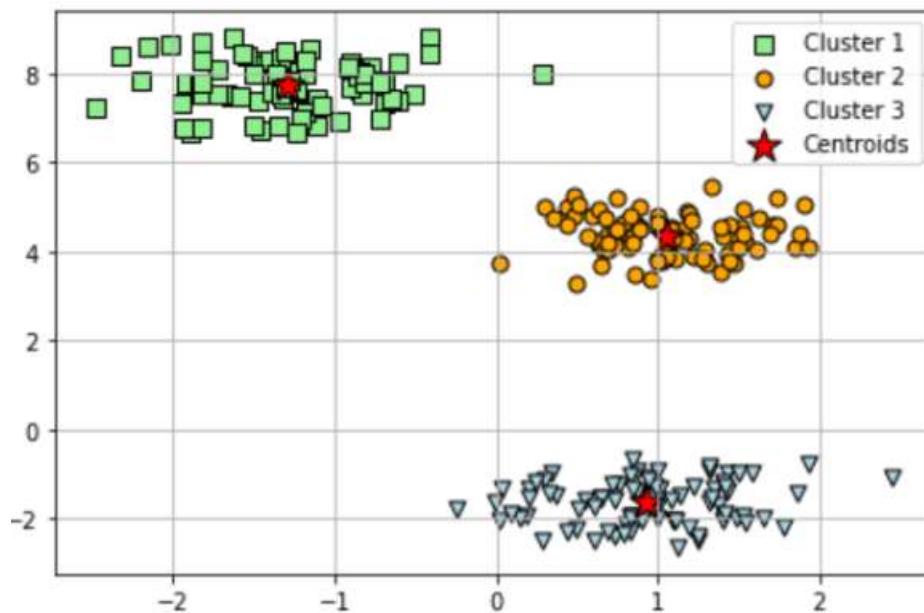


Fig: Cluster

Reference: https://en.wikipedia.org/wiki/Cluster_analysis

3.2.2 Why Clustering?

When you are working with large datasets, an efficient way to analyze them is to first divide the data into logical groupings, aka clusters. This way, you could extract value from a large set of unstructured data. It helps you to glance through the data to pull out some patterns or structures before going deeper into analyzing the data for specific findings.

Organizing data into clusters helps identify the data's underlying structure and finds applications across industries. For example, clustering could be used to classify diseases in the field of medical science and can also be used in customer classification in marketing research.

In some applications, data partitioning is the final goal. On the other hand, clustering is also a prerequisite to preparing for other artificial intelligence or machine learning problems. It is an efficient technique for knowledge discovery in data in the form of recurring patterns, underlying rules, and more

3.2.3 Types of Clustering Methods/Algorithms

Clustering helps in performing surface-level analyses of the unstructured data. The cluster formation depends upon different parameters like shortest distance, graphs, and density of the data points. Grouping into clusters is conducted by finding the measure of similarity between the objects based on some metric called the similarity measure. It is easier to find similarity measures in a lesser number of features.

Creating similarity measures becomes a complex process as the number of features increases. Different types of clustering approaches in data mining use different methods to group the data from the datasets. This section describes the clustering approaches.

Connectivity-based Clustering (Hierarchical Clustering)

Hierarchical clustering, also known as connectivity-based clustering, is based on the principle that every object is connected to its neighbours depending on their proximity distance (degree of relationship). The clusters are represented in extensive hierarchical structures separated by a maximum distance required to connect the cluster parts. Agglomerative clustering and Divisive clustering are the two main approaches in Hierarchical clustering.

Centroid-based or Partition Clustering

Centroid-based clustering is the easiest of all the clustering types in data mining. It works on the closeness of the data points to the chosen central value. The datasets are divided into a given number of clusters, and a vector of values references every cluster. The input data variable is compared to the vector value and enters the cluster with minimal difference.

Pre-defining the number of clusters at the initial stage is the most crucial yet most complicated stage for the clustering approach. Despite the drawback, it is a vastly used clustering approach for surfacing and optimizing large datasets. **The K-Means algorithm lies in this category.**

These groups of clustering methods iteratively measure the distance between the clusters and the characteristic centroids using various distance metrics. These are either Euclidian distance, Manhattan Distance or Minkowski Distance.

The major setback here is that we should either intuitively or scientifically (Elbow Method) define the number of clusters, “k”, to begin the iteration of any clustering machine learning algorithm to start assigning the data points.

Also, owing to their simplicity in implementation and also interpretation, these algorithms have wide application areas viz., market segmentation, customer segmentation, text topic retrieval, image segmentation, etc.

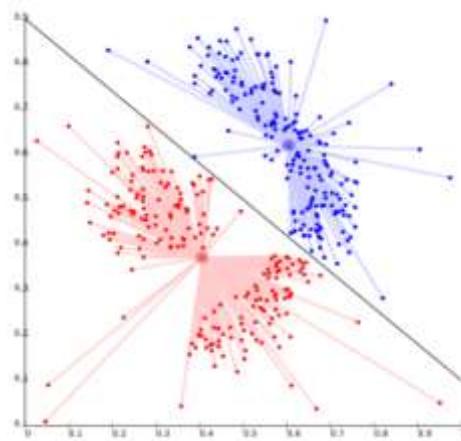


Fig:Centroid Based Clustering

Reference: <https://developers.google.com/machine-learning/clustering/clustering-algorithms>

Density-based Clustering (Model-based Methods)

The first two methods discussed above depend on a distance (similarity/proximity) metric. Density-based clustering method considers density ahead of distance. Data is clustered by regions of high concentrations of data objects bounded by areas of low concentrations of data objects. The clusters formed are grouped as a maximal set of connected data points.

The clusters formed vary in arbitrary shapes and sizes and contain a maximum degree of homogeneity due to similar density. This clustering approach includes the noise and outliers in the datasets effectively.

When performing most of the clustering, we take two major assumptions: the data is devoid of any noise and the shape of the cluster so formed is purely geometrical (circular or elliptical).

DBSCAN and OPTICS are the two most common examples of density models.

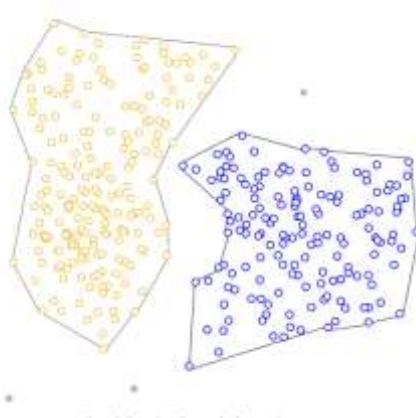


Fig: Density based Clustering

Reference: <https://developers.google.com/machine-learning/clustering/clustering-algorithms>

Distribution-Based Clustering

Until now, the clustering techniques as we know them are based on either proximity (similarity/distance) or composition (density). There is a family of clustering algorithms that take a totally different metric into consideration – probability.

Distribution-based clustering creates and groups data points based on their likely hood of belonging to the same probability distribution (Gaussian, Binomial, etc.) in the data.

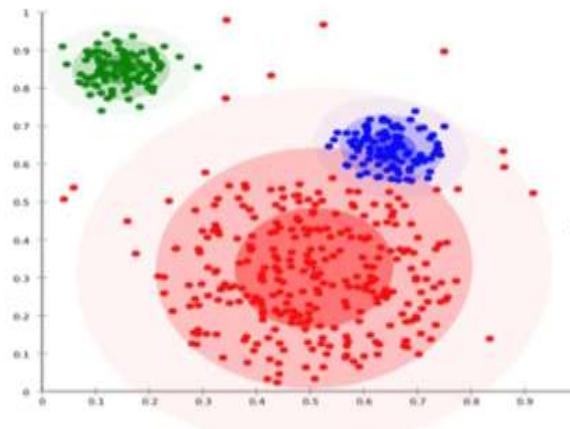


Fig: distribution-based clustering.

Reference: [Types of Clustering Algorithms in Machine Learning With Examples \(analytixlabs.co.in\)](https://www.analytixlabs.co.in/types-of-clustering-algorithms-in-machine-learning-with-examples/)

It is a probability-based distribution that uses statistical distributions to cluster the data objects. The cluster includes data objects that have a higher probability to be in it. Each cluster has a central point, the higher the distance of the data point from the central point, the lesser will be its probability to get included in the cluster.

A major drawback of density and boundary-based approaches is in specifying the clusters apriori to some of the algorithms and mostly the definition of the shape of the clusters for most of the algorithms. There is at least one tuning or hyper-parameter which needs to be selected and not only that is trivial but also any inconsistency in that would lead to unwanted results.

Distribution-based clustering has a vivid advantage over the proximity and centroid-based clustering methods in terms of flexibility, correctness, and shape of the clusters formed. The major problem however is that these clustering methods work well only with synthetic or simulated data or with data where most of the data points most certainly belong to a predefined distribution, if not, the results will overfit.

3.3 Various Distance Metrics

Distance metrics play an important role in machine learning. They provide a strong foundation for several machine learning algorithms like k-nearest neighbors for supervised learning and k-means clustering for unsupervised learning. Different

distance metrics are chosen depending upon the type of the data. So, it is important to know the various distance metrics and the intuitions behind it.

An effective distance metric improves the performance of our machine learning model, whether that's for classification tasks or clustering.

There are several measures of distance that can be used, and it is important to be aware of them while considering the best solution for a given situation to avoid errors and interpretation issues.

Types of Distance Metrics in Machine Learning

- Euclidean Distance
- Manhattan Distance
- Minkowski Distance
- Hamming Distance
- Cosine Similarity
- Mahalanobis Distance
- Chebyshev distance

Euclidean distance, Manhattan distance, Minkowski distance and hamming distance are already explained in Supervised Machine Learning. We will discuss Cosine similarity , Mahalanobis distance and chebyshev distance in details before understanding concept of K means clustering.

Cosine Distance / Similarity

Cosine similarity is a measure of similarity between two non-zero vectors. It is calculated as an inner product of the two vectors that measures the cosine of the angle between them. The vectors which are most similar will have 0 degree between them. In other words, the most similar vectors will coincide with each other. The value of $\cos 0$ is 1. The vector which will be opposite to each other or most dissimilar will have value as -1 ($\cos(180\text{deg})$).

The advantage of using cosine distance is the sheer speed at calculating distances between sparse vectors. For instance, if there are 500 attributes collected about houses and 200 of these were mutually exclusive (meaning that one house had them but the others don't), then there would only be need to include 300 dimensions in the calculation.

Here is the formula for cosine similarity between two vectors a and b having attributes in n dimensions

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \dots + a_n^2}$$

$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \dots + b_n^2}$$

Fig: Cosine similarity

Reference: <https://vitalflux.com/different-types-of-distance-measures-in-machine-learning/>

The diagram below represents cosine similarity between two vectors having different angle between them

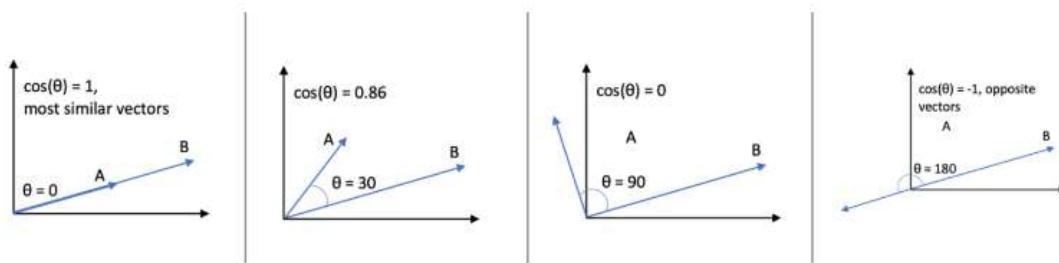


Fig: Cosine similarity b/w two vectors

Reference: <https://vitalflux.com/different-types-of-distance-measures-in-machine-learning/>

Mahalanobis distance

Mahalanobis distance is one type of statistical distance measure which is used to compute the distance from the point to the centre of a distribution. It is ideal to solve the outlier detection problem. The distance of a point P from probability distribution D is how far away standard deviation P is from the mean of probability distribution D. If the point P is at the mean of the probability distribution, the distance is zero (0).

Two points may seem to have same Euclidean distance but different Mahalanobis distance and hence are not similar. Look at the diagram given below.

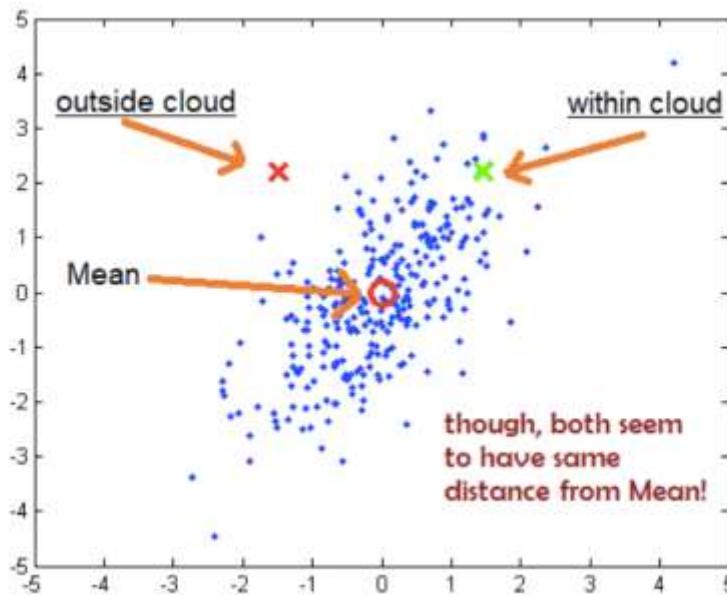


Fig: Mahalanobis distance (Point X is found as outlier)

Reference: <https://vitalflux.com/different-types-of-distance-measures-in-machine-learning/>

You may note that point X can be seen as an outlier in the data distribution shown in the above diagram although the Euclidean distance of red and green points is same from the mean.

Chebyshev Distance

Chebyshev distance is defined as the greatest of difference between two vectors along any coordinate dimension. In other words, it is simply the maximum distance along one axis. Due to its nature, it is often referred to as Chessboard distance since the minimum number of moves needed by a king to go from one square to another is equal to Chebyshev distance.

$$D(x, y) = \max_i (|x_i - y_i|)$$

Reference: <https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa>

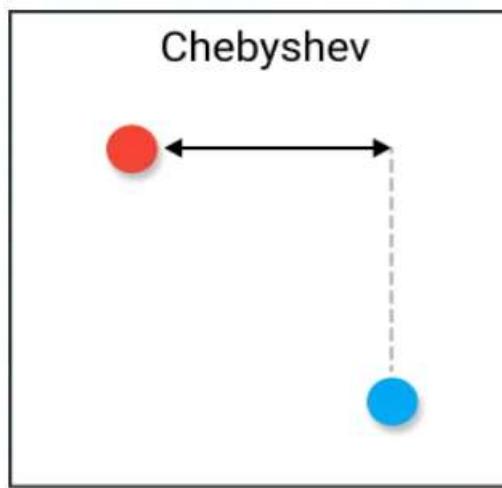


Fig: Chebyshev distance

Reference: <https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa>

Chebyshev is typically used in very specific use-cases, which makes it difficult to use as an all-purpose distance metric, like Euclidean distance or Cosine similarity. For that reason, it is suggested to only use it when you are absolutely sure it suits your use-case.

As mentioned before, Chebyshev distance can be used to extract the minimum number of moves needed to go from one square to another. Moreover, it can be a useful measure in games that allow unrestricted 8-way movement. In practice, Chebyshev distance is often used in warehouse logistics as it closely resembles the time an overhead crane takes to move an object. This is how we can calculate the distance between datapoints, which is the core concept behind our next algorithm called K-Means Clustering.

3.4 K Means Clustering

K-Means Clustering is an unsupervised machine learning algorithm. In contrast to traditional supervised machine learning algorithms, K-Means attempts to classify data without having first been trained with labeled data. Once the algorithm has been run and the groups are defined, any new data can be easily assigned to the most relevant group.

The real-world applications of K-Means include:

- 1 customer profiling
- 2 market segmentation
- 3 computer vision
- 4 search engines
- 5 astronomy

3.4.1 Main Idea of K means Clustering

The idea behind K-means clustering is to divide a dataset into a specified number of clusters (k), where all the points within the same cluster are similar to one another, and those in different clusters are different.

It starts by randomly assigning each data point to a cluster, and then it iteratively improves the clusters by moving the data points to the cluster center that is closest to them. This logic continues until the cluster assignments stop changing, or a maximum number of iterations is reached.

3.4.2 How Does the K-means Algorithm Work?

The working of the K-Means clustering in machine learning is explained in the below steps:

Step 1: First, decide the number of clusters, i.e., K .

Step 2: Select random K points or centroids. The centroids may not be from the input dataset.

Step 3: Assign each data point to its closest centroid. It will form the predefined K clusters.

Step 4: Calculate a new centroid of each cluster, taking an average of samples belonging to the same cluster.

Step 5: Repeat step 3, which means reassigning each datapoint to the new closest centroid of each cluster.

Step 6: If no new reassignment occurs, then the model is ready. Else, go to step 4.

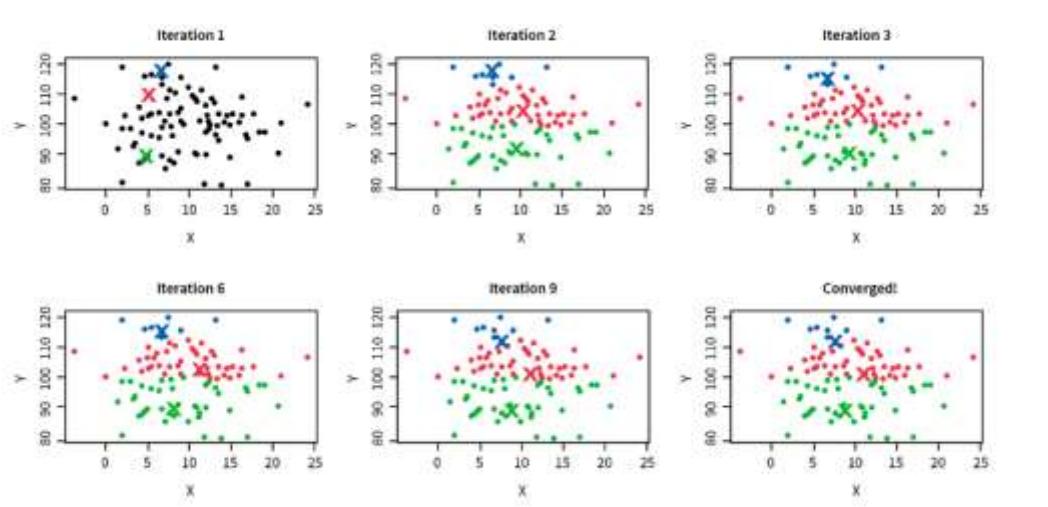


Fig: Steps in K-means clustering

Reference: <https://www.scaler.com/topics/machine-learning/k-means-clustering-in-machine-learning/>

How to Choose the Value of "K number of clusters" in K-Means Clustering?

Although there are many choices available for choosing the optimal number of clusters, the Elbow Method is one of the most popular and appropriate methods. The Elbow Method uses the idea of WCSS value, which is short for Within Cluster Sum of Squares. Within the sum of squares (WSS) is defined as the sum of the squared distance between each member of the cluster and its centroid.

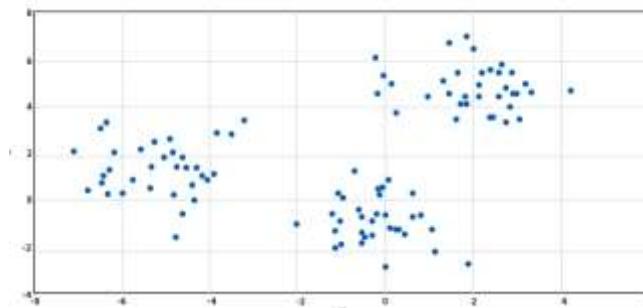
$$WSS = \sum_{i=1}^m (x_i - c_i)^2$$

Where x_i = data point and c_i = closest point to centroid

Fig: WSS formula

Reference: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/k-means-clustering-algorithm>

The WSS is measured for each value of K. The value of K, which has the least amount of WSS, is taken as the optimum value.



The above figure clearly shows that the points' distribution is forming 3 clusters. Now, let's see the plot for the loss function for different values of K. WSS is on the y-axis and number of clusters on the x-axis.

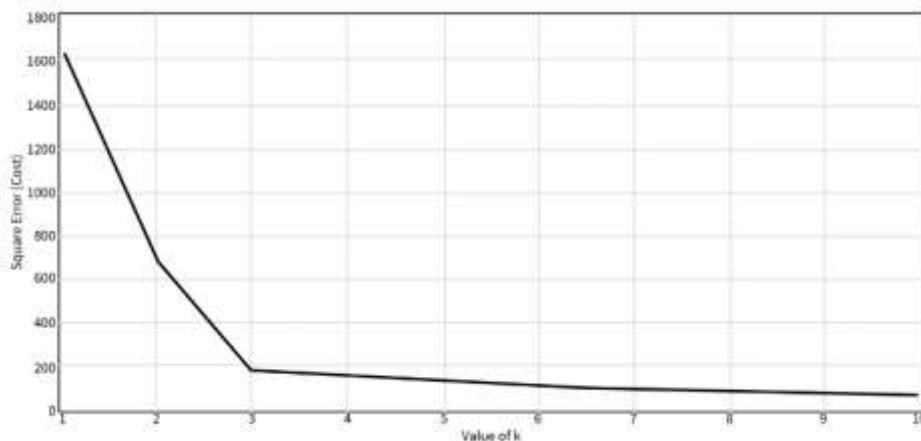


Fig: loss function for different values of K

Reference: <https://www.scaler.com/topics/machine-learning/k-means-clustering-in-machine-learning/>

You can see that there is a very gradual change in the value of WSS as the K value increases from 2. The elbow is forming at K=3. So the optimal value will be 3 for performing K-Means.

3.5 Practical: Implementation of K-Means Clustering using Python-Scikit Learn Library

PRACTICAL: We will also work on dataset for Customer Segmentation. This will help Managers of Mall to market the things effectively.

Customer segmentation is the practice of dividing a company's customers into groups that reflect similarity among customers in each group. The goal of segmenting customers is to decide how to relate to customers in each segment in order to maximize the value of each customer to the business.

The Link for the project is given below:

3.6 The Silhouette method

Silhouette score is used to evaluate the quality of clusters created using clustering algorithms such as K-Means in terms of how well samples are clustered with other samples that are similar to each other. The Silhouette score is calculated for each

sample of different clusters. To calculate the Silhouette score for each observation/data point, the following distances need to be found out for each observations belonging to all the clusters:

- Mean distance between the observation and all other data points in the same cluster. This distance can also be called a mean intra-cluster distance. The mean distance is denoted by a
- Mean distance between the observation and all other data points of the next nearest cluster. This distance can also be called a mean nearest-cluster distance. The mean distance is denoted by b.

The Silhouette Coefficient for a sample is:

$$S = \frac{(b-a)}{\max(a,b)}.$$

Fig: Silhouette Coefficient

Reference: <https://vitalflux.com/kmeans-silhouette-score-explained-with-python-example/>

The value of the Silhouette score varies from -1 to 1. If the score is 1, the cluster is dense and well-separated than other clusters. A value near 0 represents overlapping clusters with samples very close to the decision boundary of the neighbouring clusters. A negative score [-1, 0] indicates that the samples might have got assigned to the wrong clusters.

Finding Best Value of K Using Silhouette Plot

We will use YellowBrick - a machine learning visualization library to draw the silhouette plots and to perform comparative analysis. You can install yellowbrick using pip install yellowbrick statement.

We will use Sklearn IRIS dataset.

```
from sklearn import datasets
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Load IRIS dataset
iris = datasets.load_iris()

X = iris.data
y = iris.target

# Instantiate the KMeans models
km = KMeans(n_clusters=3, random_state=42)

# Fit the KMeans model.
km.fit_predict(X)
```

```

from sklearn.metrics import silhouette_samples, silhouette_score

# Calculate Silhouette Score
score = silhouette_score(X, km.labels_, metric='euclidean')

# Print the score
print('Silhouette Score: %.3f' % score)

Silhouette Score: 0.553

```

```

from yellowbrick.cluster import SilhouetteVisualizer

fig, ax = plt.subplots(2, 2, figsize=(15,8))
for i in [2, 3, 4, 5]:
    ...
    Create KMeans instance for different number of clusters
    ...
    km = KMeans(n_clusters=i, init='k-means++', n_init=10, max_iter=100, random_state=42)
    q, mod = divmod(i, 2)
    ...
    Create SilhouetteVisualizer instance with KMeans instance
    Fit the visualizer
    ...
    visualizer = SilhouetteVisualizer(km, colors='yellowbrick', ax=ax[q-1][mod])
    visualizer.fit(X)

```

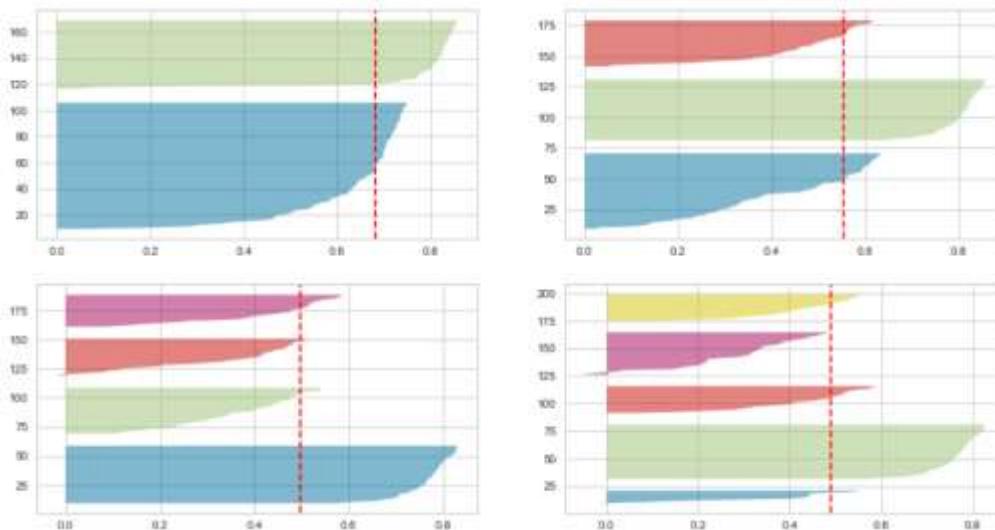


Fig: Silhouette plots for 2, 3, 4, and 5 clusters

Reference: <https://vitalflux.com/kmeans-silhouette-score-explained-with-python-example/>

Analysis on Silhouette plots

The value of n_clusters as 4 and 5 looks to be suboptimal for the given data due to the following reasons:

- Presence of clusters with below-average silhouette scores
- Wide fluctuations in the size of the silhouette plots.

The value of 2 and 3 for n_clusters looks to be the optimal one.

- The silhouette score for each cluster is above average silhouette scores. Also, the fluctuation in size is similar. The thickness of the silhouette plot representing each cluster also is a deciding point.
- For the plot with n_cluster 3 (top right), the thickness is more uniform than the plot with n_cluster as 2 (top left) with one cluster thickness much more than the other. Thus, one can select the optimal number of clusters as 3

PRACTICAL: We will use Sklearn inbuilt IRIS dataset. We will perform silhouette analysis to evaluate clusters formed by k-means clustering algorithm.

The Link for the project is given below:

Advantages of K Means Clustering

- Simple and easy to implement: The k-means algorithm is easy to understand and implement, making it a popular choice for clustering tasks.
- Fast and efficient: K-means is computationally efficient and can handle large datasets with high dimensionality.
- Scalability: K-means can handle large datasets with a large number of data points and can be easily scaled to handle even larger datasets.
- Flexibility: K-means can be easily adapted to different applications and can be used with different distance metrics and initialization methods.

Disadvantages of K Means Clustering

- Requires specifying the number of clusters: The number of clusters k needs to be specified before running the algorithm, which can be challenging in some applications.
- Sensitive to outliers: K-means is sensitive to outliers, which can have a significant impact on the resulting clusters.

3.7 Hierarchical Clustering Concept

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster. It creates groups so that objects within a group are similar to each other and different from objects in other groups. Clusters are visually represented in a hierarchical tree called a dendrogram.

Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm.

An Example of Hierarchical Clustering: Let's consider that we have a set of cars and we want to group similar ones together. Look at the image shown below:

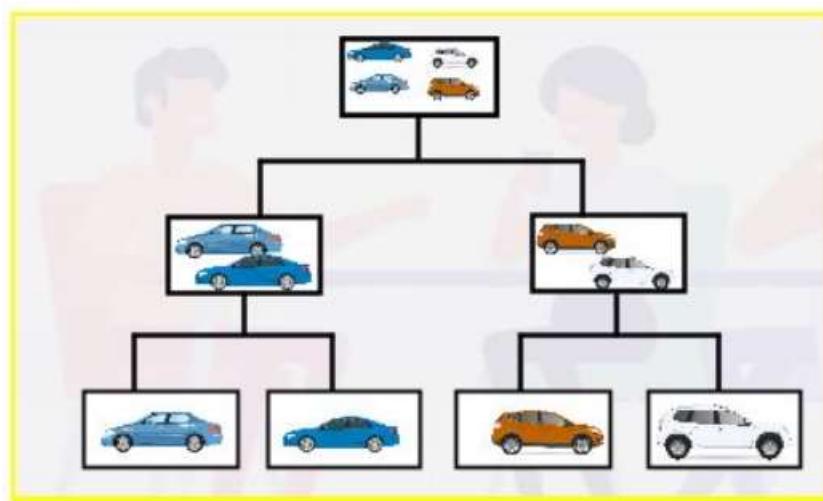


Fig: Set of Cars

Reference: <https://www.simplilearn.com/tutorials/data-science-tutorial/hierarchical-clustering-in-r>

For starters, we have four cars that we can put into two clusters of car types: sedan and SUV. Next, we'll bunch the sedans and the SUVs together. For the last step, we can group everything into one cluster and finish when we're left with only one cluster.

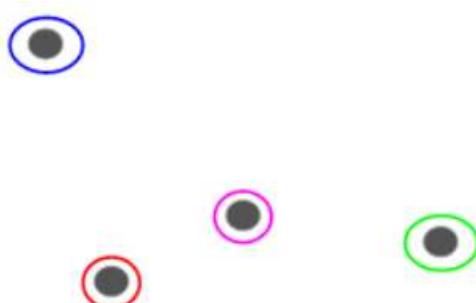
3.7.1 Types of Hierarchical Clustering

Hierarchical clustering is divided into:

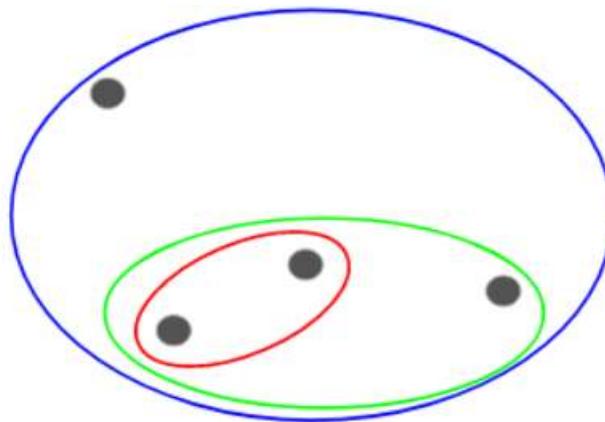
- Agglomerative
- Divisive

Agglomerative Hierarchical Clustering is popularly known as a bottom-up approach, where in each data or observation is treated as its cluster. A pair of clusters are combined until all clusters are merged into one big cluster that contains all the data.

Suppose there are 4 data points. We will assign each of these points to a cluster and hence will have 4 clusters in the beginning



Now, based on the similarity of these clusters, we can combine the most similar clusters together and repeat this process until only a single cluster is left:



We are merging (or adding) the clusters at each step, right? Hence, this type of clustering is also known as additive hierarchical clustering.

Steps for Agglomerative Clustering

The steps for agglomerative clustering are as follows:

- Start assigning each observation as a single point cluster, so that if we have N observations, we have N clusters, each containing just one observation.
- Compute the proximity matrix using a distance metric.
- Use a linkage function to group objects into a hierarchical cluster tree based on the computed distance matrix from the above step.
- Data points with close proximity are merged together to form a cluster.
- Repeat steps 3 and 4 until a single cluster remains.

Proximity Matrix

The proximity matrix is a matrix consisting of the distance between each pair of data points. The distance is computed by a distance function. Euclidean distance is one of the most commonly used distance functions.

	x_1	x_2	x_3	...	x_n
x_1	$d(x_1, x_1)$	$d(x_1, x_2)$	$d(x_1, x_3)$...	$d(x_1, x_n)$
x_2	$d(x_2, x_1)$	$d(x_2, x_2)$	$d(x_2, x_3)$...	$d(x_2, x_n)$
x_3	$d(x_3, x_1)$	$d(x_3, x_2)$	$d(x_3, x_3)$...	$d(x_3, x_n)$
...
x_n	$d(x_n, x_1)$	$d(x_n, x_2)$	$d(x_n, x_3)$		$d(x_n, x_n)$

Fig: Proximity matrix of n points

Reference: <https://builtin.com/machine-learning/agglomerative-clustering>

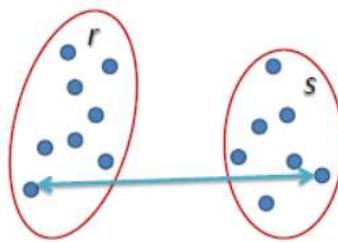
The above proximity matrix consists of n points named x, and the $d(x_i, x_j)$ represents the distance between the points.

In order to group the data points in a cluster, a linkage function is used where the values in the proximity matrix are taken and the data points are grouped based on similarity. The newly formed clusters are linked to each other until it forms a single cluster containing all the data points.

Linkage Methods

The most common linkage methods are as follows:

Complete linkage: In complete linkage hierarchical clustering, the distance between two clusters is defined as the longest distance between two points in each cluster. For example, the distance between clusters “r” and “s” to the left is equal to the length of the arrow between their two furthest points.

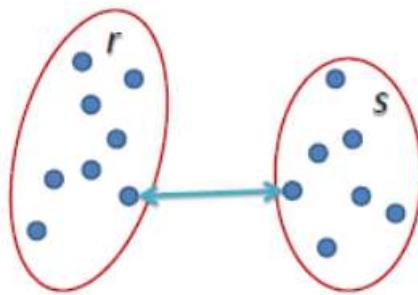


$$L(r, s) = \max(D(x_{ri}, x_{sj}))$$

Fig: Complete linkage

Reference: https://www.saedsayad.com/clustering_hierarchical.htm

Single linkage: In single linkage hierarchical clustering, the distance between two clusters is defined as the shortest distance between two points in each cluster. For example, the distance between clusters “r” and “s” to the left is equal to the length of the arrow between their two closest points.

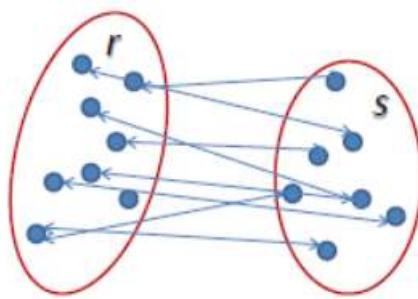


$$L(r, s) = \min(D(x_{ri}, x_{sj}))$$

Fig: Single linkage

Reference: https://www.saedsayad.com/clustering_hierarchical.htm

Average linkage: The average of all pairwise distances between elements in each pair of clusters is used to measure the distance between two clusters.



$$L(r, s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

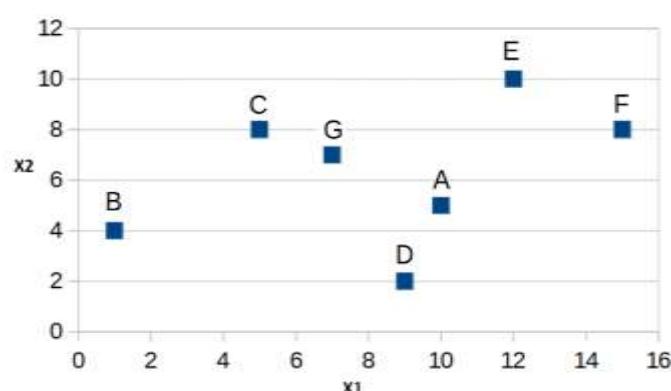
Fig: Average linkage

Reference: https://www.saedsayad.com/clustering_hierarchical.htm

Centroid linkage: Before merging, the distance between the two clusters' centroids are considered.

For Example: Clustering the following 7 data points

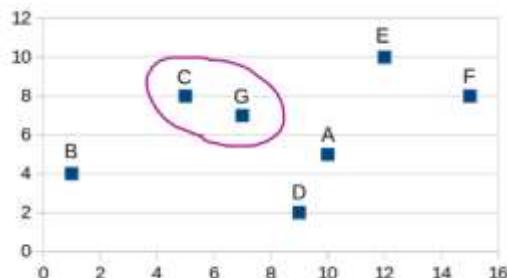
	X1	X2
A	10	5
B	1	4
C	5	8
D	9	2
E	12	10
F	15	8
G	7	7



Reference: https://www.saedsayad.com/clustering_hierarchical.htm

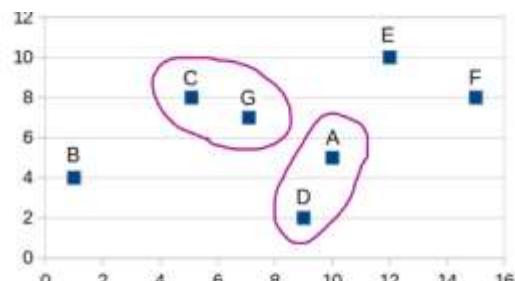
Step 1: Calculate distances between all data points using Euclidean distance function. The shortest distance is between data points C and G.

	A	B	C	D	E	F
B	9.06					
C	5.83	5.66				
D	3.16	8.25	7.21			
E	5.39	12.53	7.28	14.42		
F	5.83	14.56	10.00	16.16	3.61	
G	3.61	6.71	2.24	8.60	5.83	8.06



Step 2: We use "Average Linkage" to measure the distance between the "C,G" cluster and other data points.

	A	B	C,G	D	E
B	9.06				
C,G	4.72	6.10			
D	3.16	8.25	6.26		
E	5.39	12.53	6.50	14.42	
F	5.83	14.56	9.01	16.16	3.61



Step 3:

	A,D	B	C,G	E
B	8.51			
C,G	5.32	6.10		
E	6.96	12.53	6.50	
F	7.11	14.56	9.01	3.61

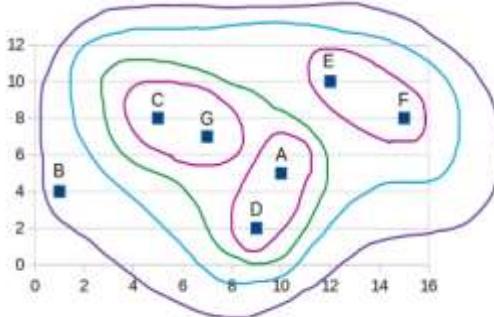


Step 4:

	A,D	B	C,G
B	8.51		
C,G	5.32	6.10	
E,F	6.80	13.46	7.65



	A,D,C,G,E,F
B	9.07



Dendrogram

A dendrogram is a tree-like diagram that records the sequences of merges or splits. More the distance of the vertical lines in the dendrogram, more the distance between those clusters.

We can set a threshold distance and draw a horizontal line (Generally, we try to set the threshold in such a way that it cuts the tallest vertical line).

Refer to the below image for clear illustration.

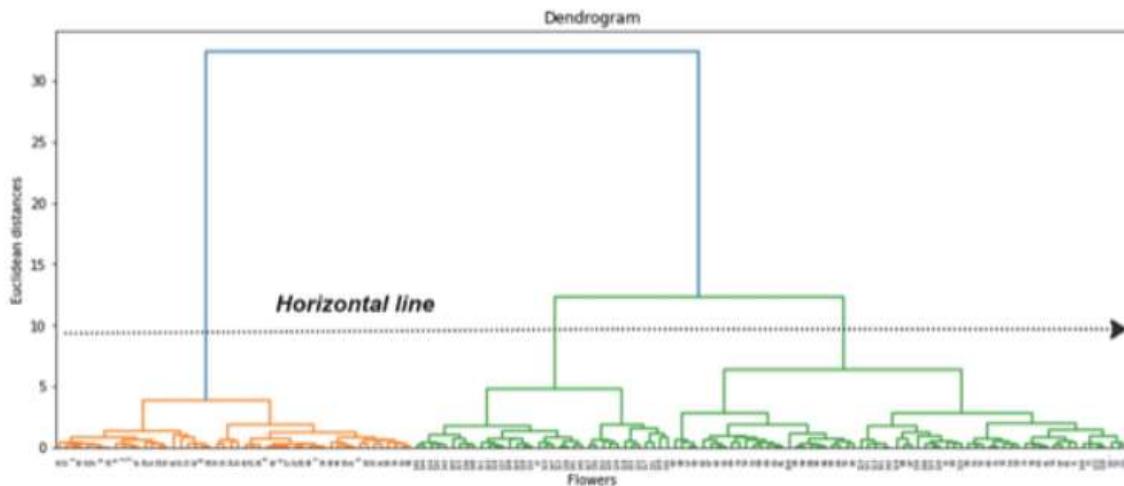


Fig: Dendrogram

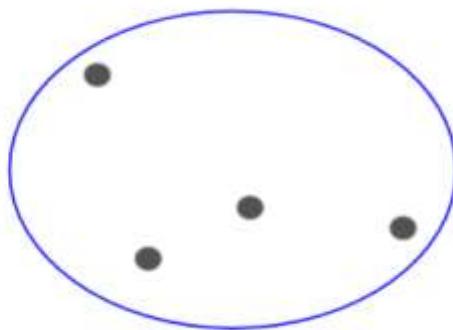
[Reference: <https://builtin.com/machine-learning/agglomerative-clustering>](https://builtin.com/machine-learning/agglomerative-clustering)

From the above figure, we have three bars below the horizontal line, so the optimal number of clusters is three.

Divisive Hierarchical Clustering

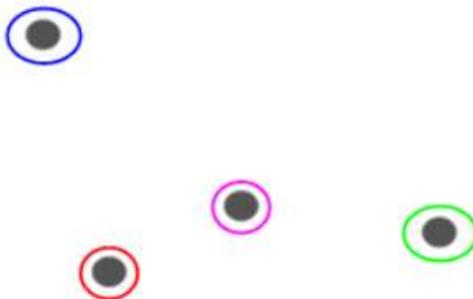
The Divisive method is the opposite of the Agglomerative method. Initially, all objects are considered in a single cluster. Then the division process is performed step by step until each object forms a different cluster. The cluster division or splitting procedure is carried out according to some principles that maximum distance between neighboring objects in the cluster. Thus, they are good at identifying large clusters. It follows a top-down approach and is more efficient than agglomerative clustering. But, due to its complexity in implementation, it doesn't have any predefined implementation in any of the major machine learning frameworks.

For Example: Consider 4 points belong to same cluster at the beginning.



[Reference: <https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering>](https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering)

Now, at each iteration, we split the farthest point in the cluster and repeat this process until each cluster only contains a single point:



Reference: <https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/>

Steps in Divisive Hierarchical Clustering

Consider all the data points as a single cluster.

- Split into clusters using any flat-clustering method, say K-Means.
- Choose the best cluster among the clusters to split further, choose the one that has the largest Sum of Squared Error (SSE).
- Repeat steps 2 and 3 until a single cluster is formed.

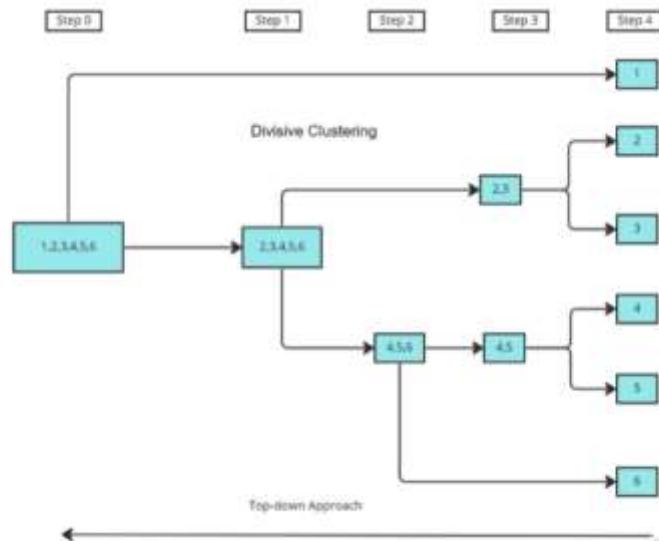


Fig: Divisive Clustering

Reference: <https://builtin.com/machine-learning/agglomerative-clustering>

In the above figure,

- The data points 1,2,3,4,5,6 are assigned to large cluster.

- After calculating the proximity matrix, based on the dissimilarity the points are split up into separate clusters.
- The proximity matrix is again computed until each point is assigned to an individual cluster.

The proximity matrix and linkage function follow the same procedure as agglomerative clustering, As the divisive clustering is not used in many places, there is no predefined class/function in any Python library.

3.8 Practical: Implementation of Hierarchical Clustering using Python- Scikit Learn Library

PRACTICAL: In this example we have Mall customer segmentation dataset. we will be interested in segmenting customers into groups based on their income, spending score etc. Business owners may use this segmentation to work with every customer segment individually and improve relationships between the company and customers or increase revenue from the specific customer category. Let's implement the Hierarchical clustering algorithm for grouping mall's customers using python:

The Link for the project is given below:

So in this section we have discussed about Hierarchical clustering and its types in details. In the next section we will explore Dimensionality reduction techniques in details.

3.9 Dimensionality Reduction

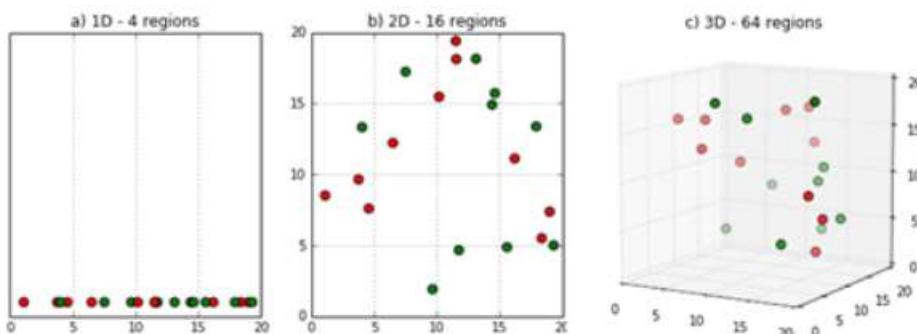
Machine learning can effectively analyze data with several dimensions. However, it becomes complex to develop relevant models as the number of dimensions significantly increases. You will get abnormal results when you try to analyze data in high-dimensional spaces. This situation refers to the curse of dimensionality in machine learning. It depicts the need for more computational efforts to process and analyze a machine-learning model.

3.9.1 What is Curse of Dimensionality?

The curse of dimensionality refers to a set of problems that arise when working with high-dimensional data and working with high dimensionality data we will have a chance of overfitting.. The dimension of a dataset corresponds to the number of attributes or features that exist in it. A dataset with a large number of attributes, generally of the order of a hundred or more, is referred to as high dimensional data. Some of the difficulties that come with high dimensional data manifest during analyzing or visualizing the data to identify patterns, and some manifest while training machine learning models.

3.9.2 Why is it a curse?

Data sparsity is an issue that arises when you go to higher dimensions. Because the amount of space represented grows so quickly that data can't keep up, it becomes sparse, as seen below.



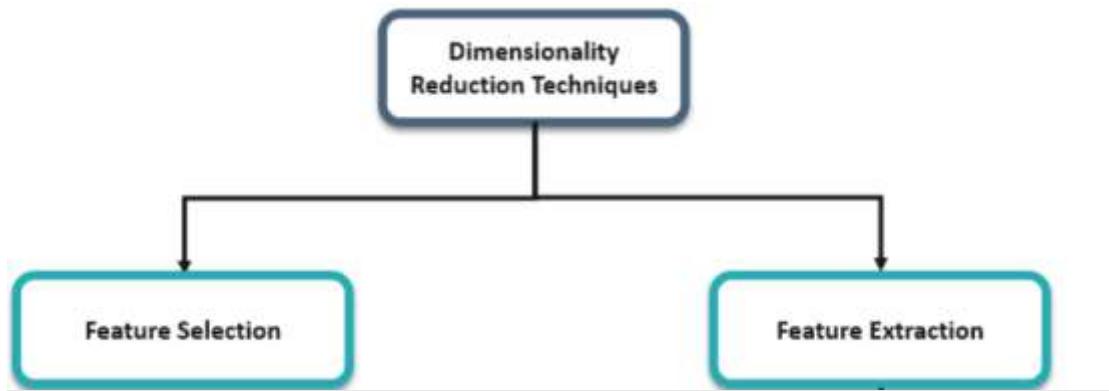
The sparsity problem is a big issue for statistical significance. As the data space approaches two dimensions and then three dimensions, the amount of data filling it decreases. As a consequence of this, the data for analysis grows dramatically.

3.9.3 Mitigating Curse of Dimensionality?

To mitigate the problems associated with high dimensional data a suite of techniques generally referred to as 'dimensional reduction techniques' are used.

Dimensionality reduction is the task of reducing the number of features in a dataset. As discussed, the higher the number of features, the more difficult it is to model them. Additionally, some of these features can be quite redundant, adding noise to the dataset and it makes no sense to include them in the training data. This is where the feature space needs to be reduced.

The process of dimensionality reduction essentially transforms data from a high-dimensional feature space to a low-dimensional feature space. Simultaneously, it is also important that meaningful properties present in the data are not lost during the transformation.



Feature Selection

Feature selection is a means of selecting the input data set's optimal, relevant features and removing irrelevant features.

Consider a table which contains information on old cars. The model decides which cars must be crushed for spare parts.

Model	Year	Miles	Owner
Jaguar E-Type 1961	1970	280000	Jenny
Porsche 911	1973	350000	Martin
BMW CSL	1972	330000	Hardik
Land Rover T	1955	6000000	Ricky

Fig: Old Car Dataset

In the above table, we can see that the model of the car, the year of manufacture, and the miles it has traveled are pretty important to find out if the car is old enough to be crushed or not. However, the name of the previous owner of the car does not decide if the car should be crushed or not. Further, it can confuse the algorithm into finding patterns between names and the other features. Hence we can drop the column.

Model	Year	Miles
Jaguar E-Type 1961	1970	280000
Porsche 911	1973	350000
BMW CSL	1972	330000

Land Rover T	1955	6000000
--------------	------	---------

Fig: Dropping columns for feature selection

Filter Method: In this method, features are dropped based on their relation to the output, or how they are correlating to the output. We use correlation to check if the features are positively or negatively correlated to the output labels and drop features accordingly. Eg: Information Gain, Chi-Square Test, Fisher's Score, etc.

Wrapper methods: This method uses the machine learning model to evaluate the performance of features fed into it. The performance determines whether it's better to keep or remove the features to improve the model's accuracy. This method is more accurate than filtering but is also more complex.

Eg: Forward Selection, Backwards Elimination, etc.

Embedded methods: The embedded process checks the machine learning model's various training iterations and evaluates each feature's importance.

This method takes care of the machine training iterative process while maintaining the computation cost to be minimum. Eg: Lasso and Ridge Regression.

Feature Extraction

Feature extraction is a dimensionality reduction technique. Unlike feature selection, which selects and retains the most significant attributes, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes.

The feature extraction technique results in a much smaller and richer set of attributes. The maximum number of features can be user-specified or determined by the algorithm. By default, the algorithm determines it.

Models built on extracted features can be of higher quality, because fewer and more meaningful attributes describe the data.

Feature extraction projects a data set with higher dimensionality onto a smaller number of dimensions. As such it is useful for data visualization, since a complex data set can be effectively visualized when it is reduced to two or three dimensions.

Due to its multiple benefits, feature extraction plays an important role in many areas, such as:

- Pattern recognition
- Image processing
- The bag of words model in Natural language processing
- Autoencoders in unsupervised learning

Here are some famous feature extraction algorithms.

- Principal Component Analysis

- Linear Discriminant Analysis
- GDA

Up until now, we have been looking in depth of Curse of Dimensionality. Here we begin looking at several unsupervised algorithms, which can highlight interesting aspects of the data without reference to any known labels.

3.10 Principal Component Analysis

Principal Component Analysis or PCA is a linear dimensionality reduction technique (algorithm) that transforms a set of correlated variables (p) into smaller k ($k \ll p$) number of uncorrelated variables called principal components while retaining as much of the variation of the original data as possible. In the context of Machine Learning (ML), PCA is an unsupervised machine learning algorithm in which we find important variables that can be useful for further regression, clustering and classification tasks.

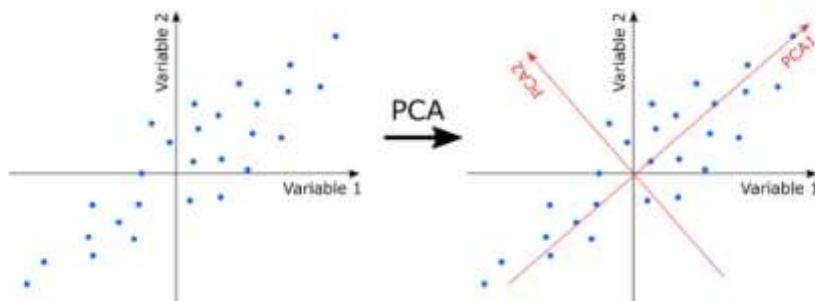


Fig: Principal Component Analysis

Reference: <https://vitalflux.com/feature-extraction-pca-python-example/>

3.10.1 What is a Principal Component?

The Principal Components are a straight line that captures most of the variance of the data. They have a direction and magnitude. Principal components are orthogonal projections (perpendicular) of data onto lower-dimensional space.

Before we delve into its inner workings, let's first get a better understanding of PCA. Principal Component Analysis example:

Let us imagine we have a dataset containing 2 different dimensions. Let the dimensions be FEATURE 1 and FEATURE 2 as tabulated below.

Feature 1	Feature 2
4	2
6	3
13	6
.....

Representation of data across Scatter Plot

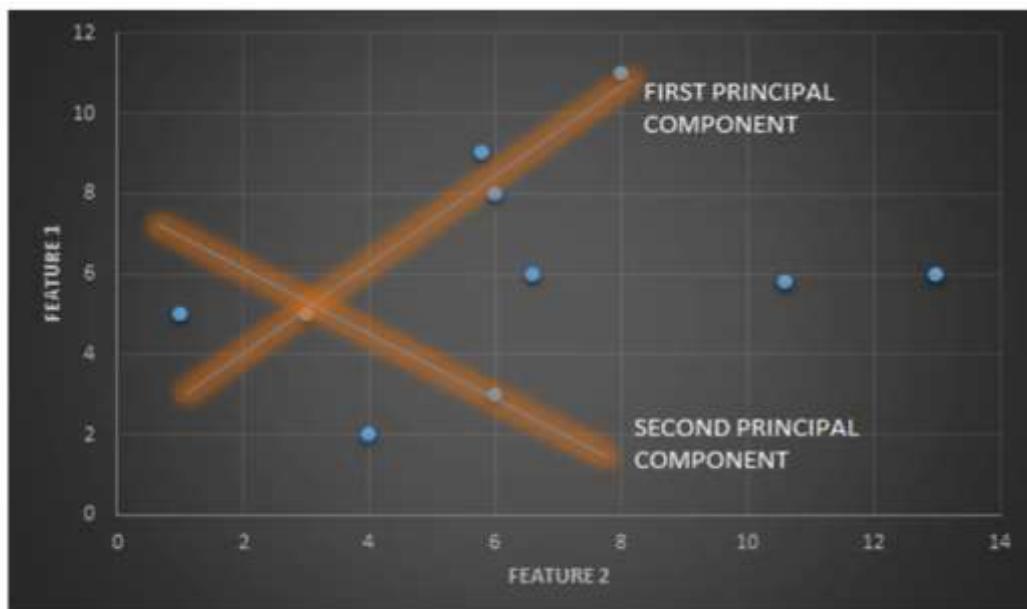
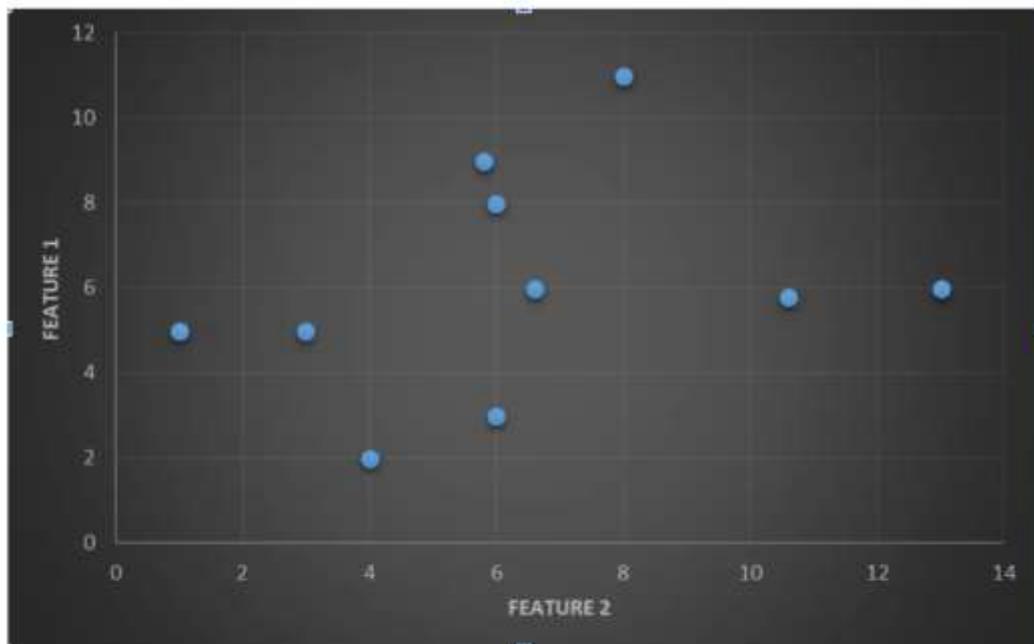


Fig: Principal Component

Reference: <https://www.turing.com/kb/guide-to-principal-component-analysis>

Here, two vector components are defined as FIRST PRINCIPAL COMPONENT (PC1) and SECOND PRINCIPAL COMPONENT (PC2) and they are computed based on a simple principle. The components that are having a similar or greater amount of variance are grouped under a single category and the components that are having varying or smaller variance are grouped under the second category.

3.10.2 Some Statistical and Mathematics key Terms in PCA

Mean

The mean (also called the average) is calculated by simply adding all the values and dividing by the number of values.

Standard Deviation

The standard deviation is a measure of how much of the data is spread from the mean. It is the square root of the variance.

Covariance

The standard deviation is calculated on a single variable. The covariance is calculated by considering two variables. The covariance becomes variance when it is calculated on the variable itself.

Covariance Matrix

We can represent all possible covariance values computed for all features of a dataset in a covariance matrix. The following image shows such a covariance matrix of a dataset of 3 variables called X, Y and Z.

$$\text{Cov} = \begin{bmatrix} \text{Cov}(X, X) & \text{Cov}(X, Y) & \text{Cov}(X, Z) \\ \text{Cov}(Y, X) & \text{Cov}(Y, Y) & \text{Cov}(Y, Z) \\ \text{Cov}(Z, X) & \text{Cov}(Z, Y) & \text{Cov}(Z, Z) \end{bmatrix}$$

Fig: Covariance matrix of 3 variable

Reference: <https://medium.com/data-science-365/statistical-and-mathematical-concepts-behind-pca-a2cb25940cd4>

$\text{Cov}(Y, X)$ is the covariance of Y with respect to X. It is the same as the $\text{Cov}(X, Y)$. The diagonal elements of the covariance matrix give you the values of variance for each variable. For example, $\text{Cov}(X, X)$ is the variance of X.

Eigenvalues and Eigenvectors

Let A be an $n \times n$ matrix. A scalar λ is called an eigenvalue of A if there is a non-zero vector x satisfying the following equation:

$$Ax = \lambda x$$

Fig: Eigen vector

Reference: <https://medium.com/data-science-365/statistical-and-mathematical-concepts-behind-pca-a2cb25940cd4>

The vector x is called the eigenvector of A corresponding to λ .

3.10.3 Steps for PCA Algorithm

1. Standardize the data: PCA requires standardized data, so the first step is to standardize the data to ensure that all variables have a mean of 0 and a standard deviation of 1.
2. Calculate the covariance matrix: The next step is to calculate the covariance matrix of the standardized data. This matrix shows how each variable is related to every other variable in the dataset.
3. Calculate the eigenvectors and eigenvalues: The eigenvectors and eigenvalues of the covariance matrix are then calculated. The eigenvectors represent the directions in which the data varies the most, while the eigenvalues represent the amount of variation along each eigenvector.
4. Choose the principal components: The principal components are the eigenvectors with the highest eigenvalues. These components represent the directions in which the data varies the most and are used to transform the original data into a lower-dimensional space.
5. Transform the data: The final step is to transform the original data into the lower-dimensional space defined by the principal components.

Applications of PCA in Machine Learning

- It is used to reduce the number of dimensions in healthcare data.
- PCA can help resize an image.
- It can be used in finance to analyze stock data and forecast returns.
- PCA helps to find patterns in the high-dimensional datasets.

Disadvantages of Principal Component Analysis

- Sometimes, PCA is difficult to interpret. In rare cases, you may feel difficult to identify the most important features even after computing the principal components. You may face some difficulties in calculating the covariances and covariance matrices.
- Sometimes, the computed principal components can be more difficult to read rather than the original set of components.

3.11 Practical: Implementation of PCA Using Python-Sklearn Library

PRACTICAL: We will use Breast Cancer dataset to perform PCA. After finding principal component we will also perform classification using logistic regression algorithm.

The link for project is given below:

PRACTICAL: In this practical we are using again Breast Cancer dataset to implement PCA using NumPy and pandas.

Link of Project:

In the next section we will explore Linear Discriminant Analysis algorithm to reduce dimensions.

3.12 Linear Discriminant Analysis (LDA) in Machine Learning

Linear discriminant analysis is an extremely popular dimensionality reduction technique used for supervised classification problems in machine learning.

Linear Discriminant Analysis was developed as early as 1936 by Ronald A. Fisher. The original Linear discriminant applied to only a 2-class problem. It was only in 1948 that C.R. Rao generalized it to apply to multi-class problems.

In this section we will discuss Linear Discriminant Analysis(LDA), the difference of LDA and PCA and related applications.

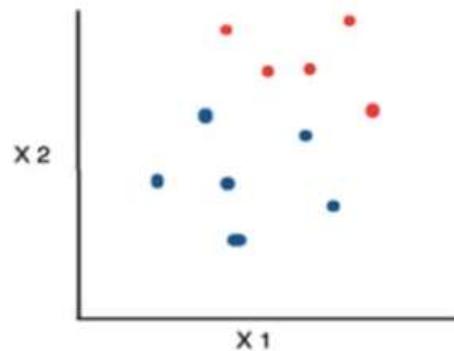
3.12.1 What is Linear Discriminant Analysis (LDA)?

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting (“curse of dimensionality”) and reduce computational costs.

LDA is similar to PCA (principal component analysis) in the sense that LDA reduces the dimensions. However, the main purpose of LDA is to find the line (or plane) that best separates data points belonging to different classes.

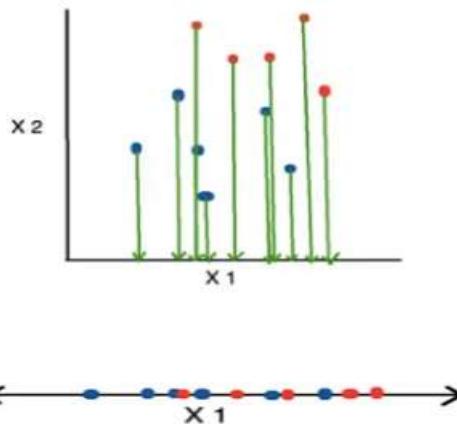
The key idea behind LDA is that the decision boundary should be chosen such that it maximizes the distance between the means of the two classes while simultaneously minimizing the variance within each classes data or within-class scatter. This criterion is known as the Fisher criterion

For example: Consider a situation where you have plotted the relationship between two variables where each color represents a different class. One is shown with a red color and the other with blue.



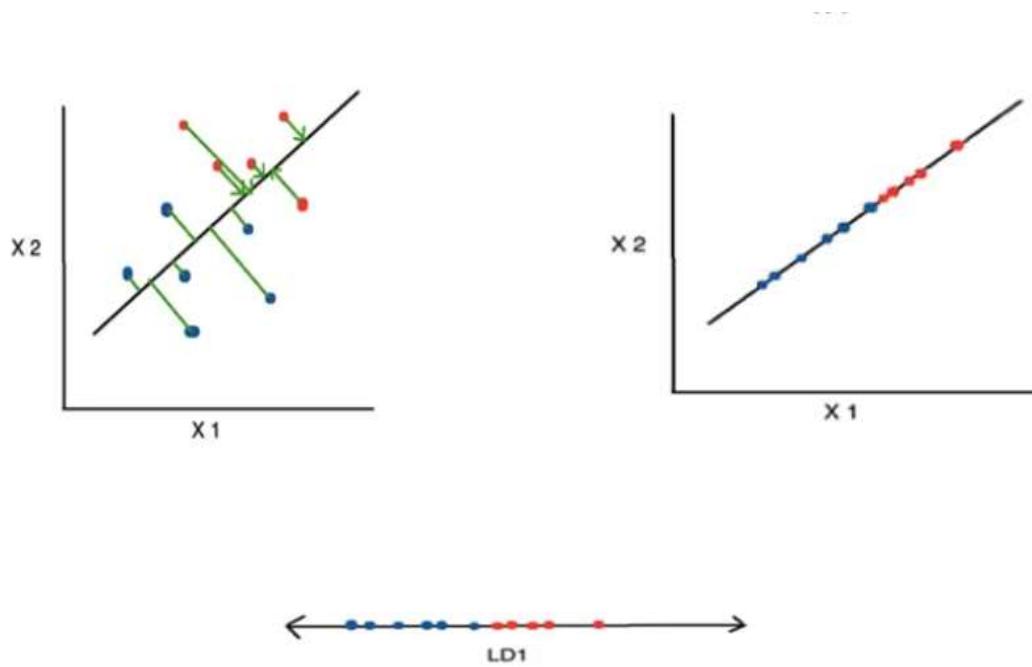
Reference: <https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning>

Suppose you want to reduce number of dimensions from 2 to 1, you can just project everything to the x-axis as shown below:



[Reference: <https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning>](https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning)

It shows overlapping. This approach neglects any helpful information provided by the second feature. However, you can use LDA to plot it. The advantage of LDA is that it uses information from both the features to create a new axis which in turn minimizes the variance and maximizes the class distance of the two variables.



[Reference: <https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning>](https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning)

In the above figure you can see that new axis increase the separation between the data points of the two classes. So, to create a new axis, Linear Discriminant Analysis uses the following criteria:

- It maximizes the distance between means of two classes.
- It minimizes the variance within the individual class.

3.12.2 Assumptions of LDA

1. Each feature (variable or dimension or attribute) in the dataset is a gaussian distribution. In other words, each feature in the dataset is shaped like a bell-shaped curve.
2. Each feature has the same variance, the value of each feature varies around the mean with the same amount on average.
3. Each feature is assumed to be randomly sampled.
4. Lack of multicollinearity in independent features. Increase in correlations between independent features and the power of prediction decreases.

3.12.3 How LDA works and steps involved

LDA algorithm works based on the following steps:

Step 1: Calculate the separability between different classes. This is also known as between-class variance and is defined as the distance between the mean of different classes.

$$S_b = \sum_{i=1}^g N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

Fig: Between class variance

Reference: <https://www.digitalvidya.com/blog/linear-discriminant-analysis/>

Step 2: Calculate the within-class variance. This is the distance between the mean and the sample of every class.

$$S_w = \sum_{i=1}^g (N_i - 1) S_i = \sum_{i=1}^g \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$$

Fig: Within class variance

Reference: <https://www.digitalvidya.com/blog/linear-discriminant-analysis/>

Step 3: Construct the lower-dimensional space that maximizes Step1 (between-class variance) and minimizes Step 2(within-class variance). In the equation below P is the lower-dimensional space projection. This is also known as Fisher's criterion.

$$P_{lda} = \arg \max_P \frac{|P^T S_b P|}{|P^T S_w P|}$$

Fig: Fisher's Criterion

Reference: <https://www.digitalvidya.com/blog/linear-discriminant-analysis/>

3.13 Practical: LDA implementation using python

PRACTCAL:

In this practical, We will use the Wine classification dataset to perform Linear Discriminant Analysis (LDA). The Wine dataset comes preloaded with Scikit-learn. The Wine dataset has 178 instances (data points). Its original dimensionality is 13 because it has 13 input features (variables). Moreover, the data points were divided into three separate classes that represent each Wine category.

The link for project is given below:

Now after implementation of LDA, we will discuss how LDA is different from PCA.

Principal Component Analysis vs. Linear Discriminant Analysis

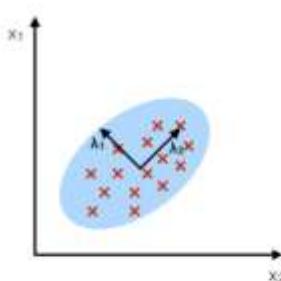
Both Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are linear transformation techniques that are commonly used for dimensionality reduction. PCA can be described as an “unsupervised” algorithm, since it “ignores” class labels and its goal is to find the directions (the so-called principal components) that maximize the variance in a dataset. In contrast to PCA, LDA is “supervised” and computes the directions (“linear discriminants”) that will represent the axes that that maximize the separation between multiple classes.

Although it might sound intuitive that LDA is superior to PCA for a multi-class classification task where the class labels are known, this might not always be the case.

For example, comparisons between classification accuracies for image recognition after using PCA or LDA show that PCA tends to outperform LDA if the number of samples per class is relatively small.

PCA:

component axes that maximize the variance



LDA:

maximizing the component axes for class-separation

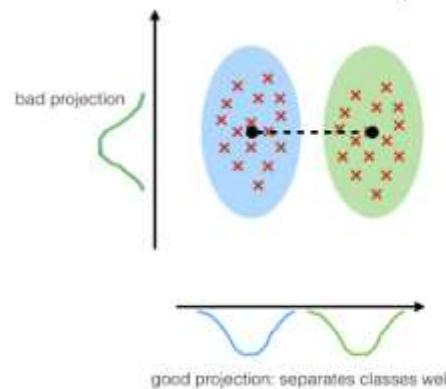


Fig: Comparison between PCA and LDA

Reference: https://sebastianraschka.com/Articles/2014_python_lda.html

Practical: In this practical, We will compare PCA with LDA using Sklearn wine dataset.

The link of project is given below:

3.14 Applications and Drawback of LDA

Real-Life Applications of LDA

Face Recognition: LDA is used in face recognition to reduce the number of attributes to a more manageable number before the actual classification. The dimensions that are generated are a linear combination of pixels that forms a template. These are called Fisher's faces.

Medical: You can use LDA to classify the patient disease as mild, moderate or severe. The classification is done upon the various parameters of the patient and his medical trajectory.

Customer Identification: You can obtain the features of customers by performing a simple question and answer survey. LDA helps in identifying and selecting which describes the properties of a group of customers who are most likely to buy a particular item in a shopping mall.

Drawbacks of Linear Discriminant Analysis (LDA)

Although, LDA is specifically used to solve supervised classification problems for two or more classes which are not possible using logistic regression in machine learning. But LDA also fails in some cases where the Mean of the distributions is shared. In this case, LDA fails to create a new axis that makes both the classes linearly separable.

To overcome such problems, we use non-linear Discriminant analysis in machine learning.

3.15 Kernel Discriminant Analysis (Generalized Discriminant Analysis)

Introduction:

Principal Component Analysis (PCA) has been widely adopted to extract abstract features and to reduce dimensionality in many pattern recognition problems. But the features extracted by PCA are actually "global" features for all pattern classes, thus they are not necessarily much representative for discriminating one class from others. Linear Discriminant Analysis (LDA), which seeks to find a linear transformation by maximising the between-class variance and minimising the within-class variance, has proved to be a suitable technique for discriminating different pattern classes. However, both the PCA and LDA are linear techniques which may be less efficient when severe non-linearity is involved. To extract the non-linear principal components, Kernel PCA (KPCA) was developed using the popular kernel technique. However, similar to the linear PCA, KPCA captures the overall variance of all patterns which are not necessary significant for discriminant purpose.

To extract the nonlinear discriminant features, Kernel Discriminant Analysis (KDA), a nonlinear discriminating method based on kernel techniques was developed. Kernel Discriminant Analysis also known as Generalized Discriminant Analysis (GDA).

Kernel Discriminant Analysis (KDA)

Kernel discriminant Analysis deals with nonlinear discriminant analysis using kernel function operator. The underlying theory is close to the support vector machines (SVM) insofar as the GDA/KDA method provides a mapping of the input vectors into high-dimensional feature space. Similar to LDA, the objective of KDA is to find a projection for the features into a lower-dimensional space by maximizing the ratio of between-class scatters to within-class scatter. The main idea is to map the input space into a convenient feature space in which variables are nonlinearly related to the input space.

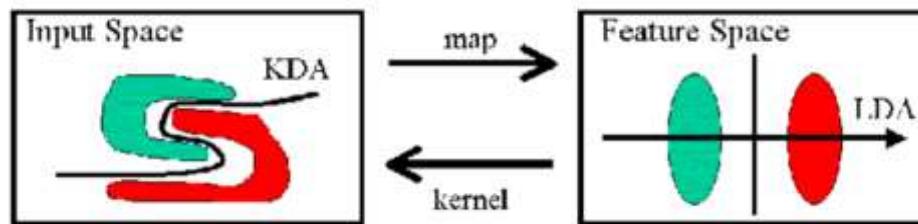


Fig: Kernel Discriminant Analysis

Reference:https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/LI1/kda/index.html

The principle of KDA can be illustrated in above image. Owing to the severe non-linearity, it is difficult to directly compute the discriminating features between the two classes of patterns in the original input space (left). By defining a non-linear mapping from the input space to a high-dimensional feature space (right), we (expect to) obtain a linearly separable distribution in the feature space. Then LDA, the linear technique, can be performed in the feature space to extract the most significant discriminating features.

KDA uses a kernel function ϕ which transform the original space X to a new high-dimensional feature space.

Steps involved in KDA

Step:1 Use Kernel function to map input training data to a high dimensional feature space, where different classes is supposed to be linearly separable.

Step:2 The Linear Discriminant Analysis (LDA) scheme is then applied to the mapped data, where it searches for those vectors that best discriminate among the classes rather than those vectors that best describe the data

Application

1. Face recognition and detection
2. Hand-written digit recognition
3. Palmprint recognition
4. Classification of malignant and benign cluster microcalcifications
5. Seed classification

Unit 4: Machine Learning with IoT

Learning Outcomes:

- Understand need of integrating Machine Learning with IoT
- Implement the connectivity of mobile sensors live data to cloud.
- Implement the connectivity of RaspberryPI machine live data to cloud.
- Learn the data acquisition from Cloud and apply Machine Learning to it.

4.1 Introduction: Machine Learning with IoT

Introduction

One of the top trending topics is machine learning and the Internet of Things. The Internet of Things produces enormous amounts of data from millions of devices. Data fuels machine learning, which derives knowledge from it.

IoT requires machine learning for analyzing enormous amounts of data with complex algorithms can aid in the decipherment of the hidden patterns in IoT data. In critical processes, machine learning inference can augment or completely replace manual processes with automated systems that use statistically derived actions.

Advantages of Integrating ML with IOT

Here we are going to discuss 5 specific advantages of AI and IoT that will change the way that businesses operate and identify.

- Enhanced Customer Relationship
- Cost-Effective
- Increased Operational Efficiency
- Highly Secured and Safe
- Focus on New Products and Services



- ENHANCED CUSTOMER RELATIONSHIP
- COST EFFECTIVE
- INCREASED OPERATIONAL EFFICIENCY
- HIGHLY SECURED & SAFE
- FOCUS ON NEW PRODUCTS & SERVICES

Fig:[IOT-ML Benefits](#)

Reference: <https://www.optisolbusiness.com/insight/top-5-benefits-of-artificial-intelligence-integration-with-iot>

Enhanced Customer Relationship

With the correct implementation, the benefit from the combination of AI and IoT is not confined only to the employees but also helps to enhance the experience of customers. Enterprises are using these emerging technologies to gather big data about customers in real-time. With the data and technologies available now, businesses can develop products and services that best fit the customer's needs. AI and IoT both play a significant role in this process. Enterprises are now automating the entire process of organizing data to make sure that the customer can receive a quick and relevant response. Customers are already getting used to receiving a quick and accurate response and reaction. Customer experience can be greatly improved because these devices can learn user preferences and adjust accordingly.

Cost Effective

AI and IoT can be a solution for saving the expenditure of the firm. AI and IoT can quickly collect and analyze data to determine if a part or procedure has become too expensive to maintain. Early access to the data allows a business to identify cost savings without sacrificing productivity. By identifying cost drivers, an organization can implement changes that lower expenses. These emerging technologies empower leaders to reduce unnecessary spending and to optimize business procedures.

Increased Operational Efficiency

Incorporating AI into IoT applications can achieve increased operational efficiency. The machine learning capabilities can process data and make predictions in ways that humans are unable to do. Large sets of data can be calculated in a short period of time with the help of this technology. Based on the calculations, recommendations can be made to make the workplace more efficient. Enterprises are already investing in these technologies to enhance productivity. AI and IoT can spot inefficiencies and recommend best practices to improve the process.

Highly Secured and Safe

The combination of both AI and IoT can provide an extra layer of security. The combination reduces workplace accidents. By pairing **machine learning** with machine-to-machine communication, enterprises can predict potential security risks and automate an immediate response. Connected sensors could be leveraged to determine potential environmental safety hazards that workers are unaware of. A safe and secure enterprise can be achieved with these solutions. Many applications pairing with IoT and AI can also help organizations to predict and well manage a variety of risks and threats such as worker safety, cyber threats, financial losses, etc.

Focus on New Products and Services

Artificial Intelligence and IoT combination can pave the way of creating new and powerful products and services. Collecting and Analyzing huge sets of helps organizations to make smart decisions based on the situations. Natural language processing (NLP) is getting better and better at letting people speak with machines, rather than requiring a human operator. AI is a natural complement to IoT deployments, enabling better offerings and operations to give a competitive edge in business performance.

4.2 Applications of ML with IoT

The integration of IoT and machine learning has a wide range of applications across various industries. The combination of these technologies allows for real-time data analysis and improved decision-making, leading to increased efficiency and cost savings. Let's take a look at how IoT machine learning is being used in the following industries.

Healthcare

In the healthcare industry, IoT machine learning can be used to monitor patients remotely and provide real-time health data to healthcare professionals. This information can be used to diagnose and treat patients more effectively, reducing the need for in-person visits and minimizing the spread of illness. IoT-powered devices such as wearable fitness trackers and smart inhalers can also provide valuable data for machine learning algorithms to analyze, helping healthcare professionals make more informed decisions.

Retail

IoT machine learning is also being utilized in the retail industry to enhance customer experiences and improve the efficiency of supply chain management. For example, retailers can use IoT sensors to track inventory levels in real-time, enabling them to make data-driven decisions about when to reorder products and minimize waste. Additionally, machine learning algorithms can be used to analyze customer purchase patterns, enabling retailers to offer personalized product recommendations and improve overall customer satisfaction.

Manufacturing

In the manufacturing industry, IoT machine learning can be used to optimize production processes, improve quality control, and reduce waste. For example, machine learning algorithms can be used to analyze data from IoT sensors on factory equipment, allowing manufacturers to identify areas for improvement and make proactive repairs before equipment breakdowns occur. This can lead to reduced downtime, improved productivity, and increased profits.

Agriculture

In the agriculture industry, IoT machine learning can be used to improve crop yields, minimize waste, and reduce the use of harmful chemicals. For example, machine learning algorithms can be used to analyze data from IoT-enabled sensors in soil and weather patterns to optimize crop irrigation and fertilizer application. This can lead to improved crop health, reduced costs, and increased profits for farmers.

Transportation and logistics

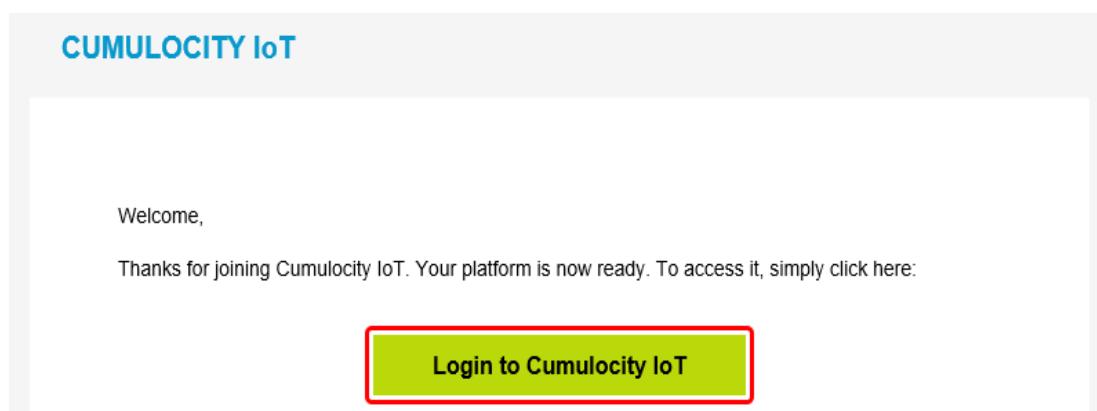
In the transportation and logistics industry, IoT machine learning can be used to improve delivery times and reduce waste. For example, machine learning algorithms can be used to analyze data from GPS-enabled vehicles to optimize delivery routes and reduce fuel consumption. This can lead to faster delivery times, reduced costs, and improved customer satisfaction.

4.3 Practical: Transferring IOT Data to Cloud Services

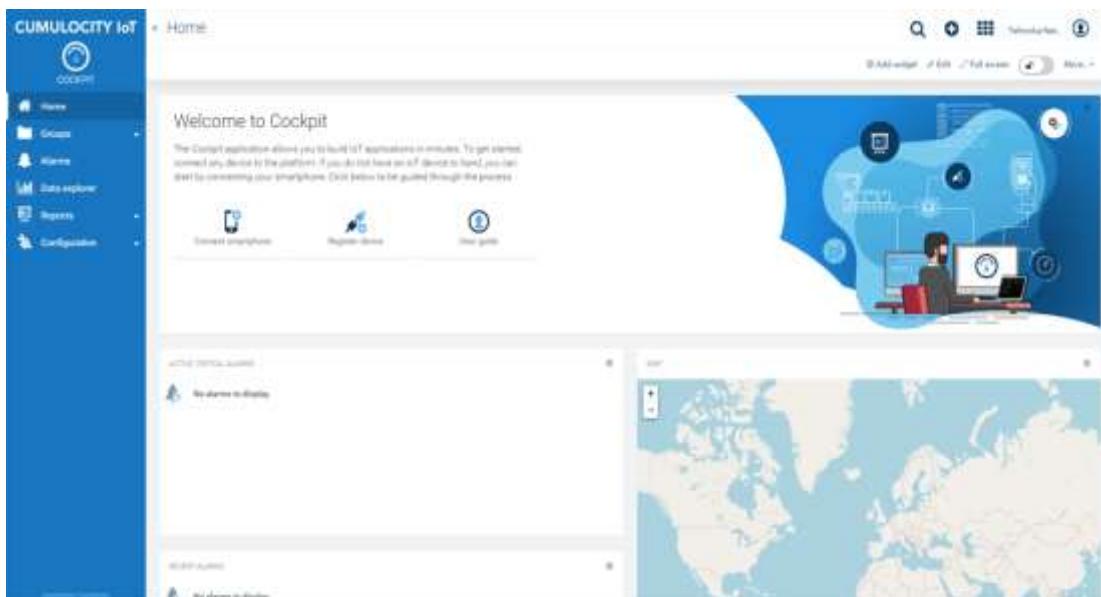
Create an account

First of all, you will need to set up your Cumulocity environment (server).

- [**Register an account**](#) on Cumulocity.
- Wait a few minutes after the registration, open your email account and look for an email with the subject 'Welcome to Cumulocity IoT'. Open it and **click 'Login to Cumulocity IoT'**.



- Upon a successful login you should be redirected to the **Cumulocity Cockpit**.

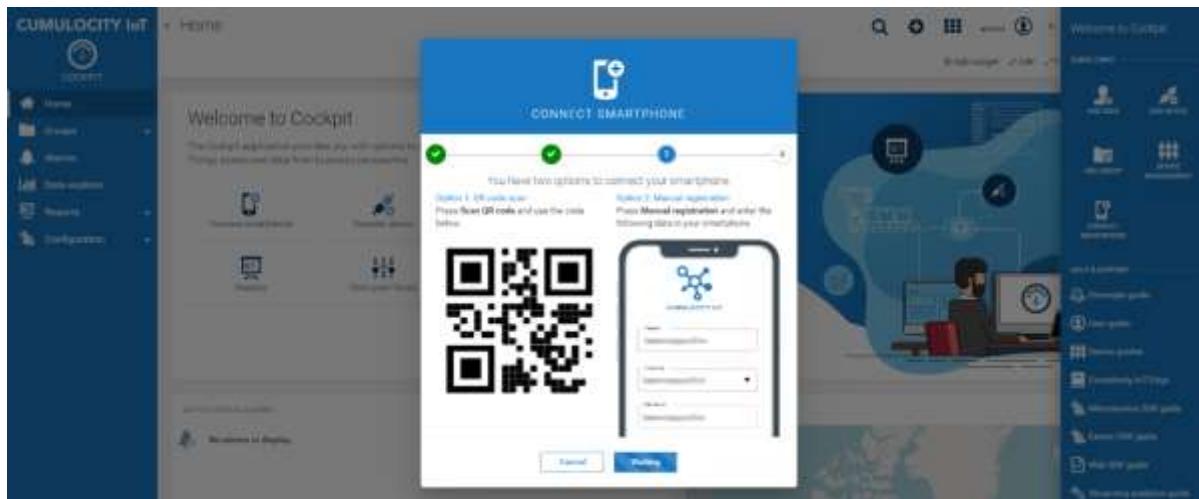


Connecting Smartphone Sensor Data to Cloud

- On a desktop or laptop computer, open a web browser and log in to your Cumulocity IoT tenant. From the Cockpit application, click **Connect Smartphone** in the right drawer or in the Welcome widget.



- Follow the instructions in the wizard to step 3, ensuring that the app is installed on the smartphone.



- From your smartphone, launch the app and tap **Connect** in the top right corner of the screen.
- Grant access to your camera if the app asks you for permission.
- Scan the QR code shown on your PC's web browser. If you can't scan the QR code, tap **Manual registration** on your smartphone and fill in the details at the right side of the wizard screen.
- Back on your smartphone, tap **Done**. Sensor measurements are sent to the server. They can be viewed in the device's dashboard.

When using the **Connect Smartphone** wizard for device registration, your smartphone is automatically registered by Cumulocity IoT and assigned to the "Phones" group. Tap **Done** on your smartphone to return to the main screen.

Registering a Raspberry PI on Cloud

Follow below steps to register a device on Cloud.

1.. Clone below repository on RaspberryPi

<https://github.com/SoftwareAG/cumulocity-devicemanagement-docker-example.git>

2.. Open repository and note down the device id

```
cat cumulocity-devicemanagement-docker-example/Agents/config/config.ini
```

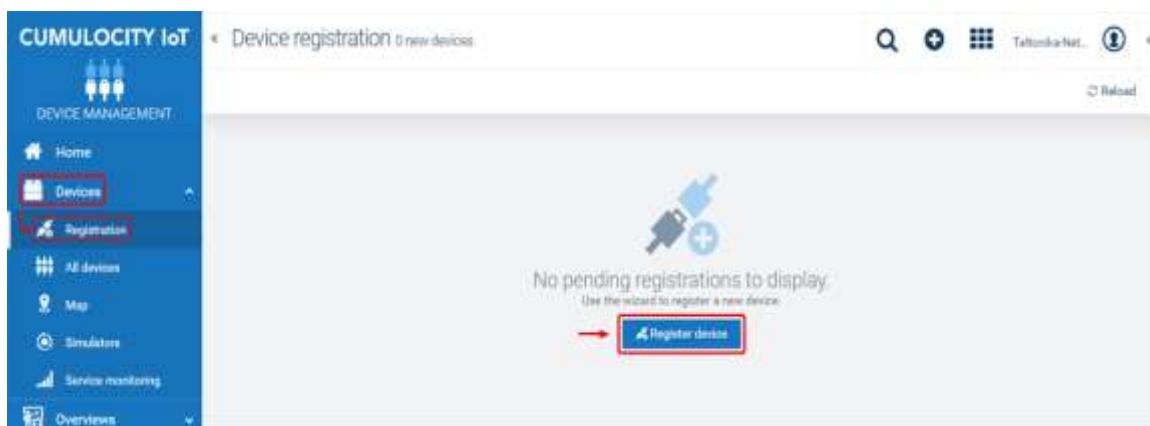
```
pi@raspberrypi:~ $ cat cumulocity-devicemanagement-docker-example/Agents/config/config.ini
[c8Y]
tenantInstance = eu-latest.cumulocity.com

[Device]
id = 08152

[Registration]
user = management/devicebootstrap
password = Fhdt1bb1f
tenant = management
tenantPostFix = /devicecontrol/deviceCredentials

[MQTT]
prefix = aggregated
prefixSignaltypes = signalType
broker = localhost
port = 1883
```

3,,Then expand the 'Devices' tab in left-hand menu, go to 'Registration' and **click 'Register device'**.



4. Enter your device's serial number into the 'Device ID' field. Then it will await for device application to be turned on

The screenshot shows the Cumulocity IoT Device Management interface. On the left, a sidebar menu includes options like Home, Devices (selected), Registration, All devices, Map, and Simulators. The main content area is titled 'Device registration' and shows a single device entry: 'myDeviceID' with status 'WAITING FOR CONNECTION'. A red 'Remove' button is present. At the bottom, it says 'Created on 23 July 2020 07:00 by admin'.

5.. Run docker in Raspberry PI using below command to connect RPI to cloud

```
sudo bash cumulocity-devicemanagement-docker-example/start.sh
```

6.. Refer back to dashboard to accept connection and now your device is connected to cloud.

4.4 Collecting Sensors Data from Cloud

1. Refer back to Cumulocity IOT Device Management dashboard

2. Open dashboard of connected devices.

The screenshot shows the 'All devices' dashboard. The sidebar menu includes Home, Devices (selected), Registration, All devices (selected), Map, Simulators, Availability, Overviews, Groups, Device-types, and Management. The main content area displays a table of 14 devices. The columns are: St..., Name, Model, Serial num..., Group, Registration ..., System ID, IMB, and Alarms. The devices listed are: Greenhouse 1 - Zone 1 CO2, Greenhouse 1, Greenhouse 1 - Zone 2 CO2, Site 1 - RII, Site 2 - RII, Gateway_09132, and 1818. Each device row has a small icon and a red 'i' icon indicating alerts.

St...	Name	Model	Serial num...	Group	Registration ...	System ID	IMB	Alarms
1	Greenhouse 1 - Zone 1 CO2				6 Nov 2020, 02:31:28	4328466		1 ●
2	Greenhouse 1				9 Nov 2020, 18:19:38	9125648		1 ●
3	Greenhouse 1 - Zone 2 CO2				16 Nov 2020, 21:17:08	7229417		1 ●
4	Site 1 - RII				25 Nov 2020, 22:08:38	1926574		1 ●
5	Site 2 - RII				25 Nov 2020, 22:09:31	6925650		1 ●
6	Gateway_09132			Phones:	18 May 2023, 22:37:28	2122328		
7	1818	vivo 1818	9c7113e4-2274-493b-b70-290f5c660969	Phones:	17 May 2023, 09:54:37	9124368		

3 Open your connected device of interest.

4.. Refer to Measurement section to view the device activity. Click on More to download device sensor data



4.5 Practical: Machine Learning on Sensor Data

The downloaded data is majorily in form of unsupervised data. Try to aggregate data from various sensors and apply Machine Learning Algorithms on them to identify the clusters and data homogeneity.

The link for project is given below:

Module II

Internet of Things

Unit 1: Internet of Things

Learning Outcomes:

- Understand the basic concept of IoT
- Understand the use and application of IoT

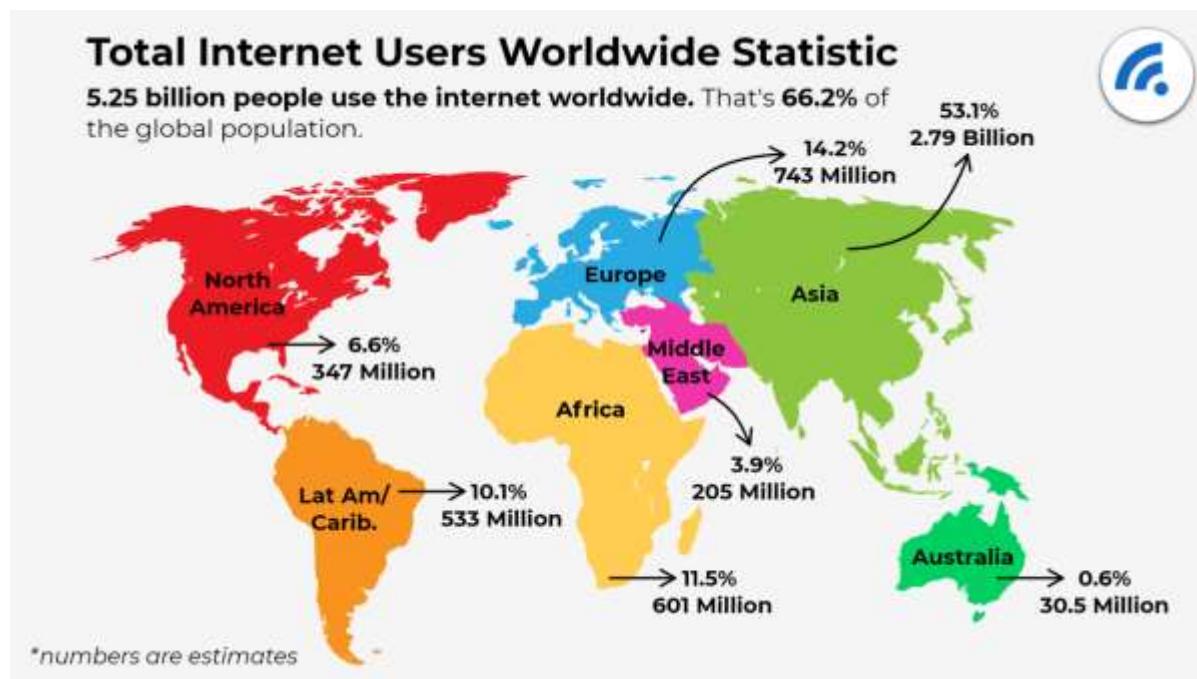
1.1 Internet Usage and Population Statistics

A total of 5 billion people around the world use the internet today – equivalent to 63 percent of the world's total population.

Internet users continue to grow too, with the latest data indicating that the world's connected population grew by almost 200 million in the 12 months to April 2022.

There are now fewer than 3 billion people who remain “unconnected” to the internet, with the majority of these people located in Southern and Eastern Asia, and in Africa.

That means that there's still plenty more work to do before the world reaches the goal of “universal access”, and the quality of people's internet access is also an important consideration.



Reference: <https://www.broadbandsearch.net/blog/internet-statistics>

However, internet users continue to grow at an annual rate of more than 4 percent, and current trends suggest that two-thirds of the world's population should be online by the middle of 2023.

What's more, the ongoing coronavirus pandemic continues to have a meaningful impact on internet user research, so actual user figures and growth rates may be higher than current data suggests.

The vast majority of the world's internet users – 92.4 percent – use a mobile phone to go online at least some of the time, and mobile phones now account for more than half of our online time, and more than half of the world's web traffic.

However, roughly two-thirds of internet users in the world's larger economies still use laptops and desktops for at least some of their online activities.

1.2 What is Internet of Things?

1.2.1 Introduction

Internet of Things (IoT) comprises things that have unique identities and are connected to the internet.

Existing devices, such as networked computers or 4G enabled mobile phones already have some form of unique identities and are also connected to the internet, the focus on IoT is in the configuration, control and networking via the internet of devices or things, that are traditionally not associated with the Internet. These include devices such as thermostats, utility meters, a blue tooth-connected headset, irrigation pumps and sensor or control circuits for an electric car's engine.

Experts forecast that by the year 2020 there will be a total of 50 billion devices/ things connected to the internet.

The scope of IoT is not limited to just connected things (Devices, appliance, machines) to the Internet. Applications on IoT networks extract and create information from lower-level data by filtering, processing, categorizing, condensing and contextualizing the data. The information obtained is then organized and structured to infer knowledge about the system and/or its user, its environment and its operations and progress towards its objectives, allowing a smarter performance.

1.2.2 Definition and characteristics of IoT

Definition

A dynamic global network infrastructure with self – configuring based on standard and interoperable communication protocols where physical and virtual “things” have identified, physical attributes, and virtual personalities and use intelligent interfaces, often communicate data associated with users and their environment Characteristics

Dynamic and self-Adapting:

IoT devices and systems may have the capability to dynamically adapt with the changing contexts and take actions based on their operating condition. Ex: Surveillance cameras can adapt their modes based on whether it is day or night.

Self – Configuring:

IoT devices may have self-Configuring capability allowing a large number of devices to work together to provide certain functionality.

Interoperable communication protocols:

IoT Devices may support a number of interoperable communication protocols and can communicate with other devices and also with the infrastructure.

Unique Identity:

Each IoT devices has a unique identity and a unique identifier.(IPaddress, URI).IoT systems may have intelligent interfaces which adapt based on the context, allow communication with users and the environment contexts.

Integrated into information network:

IoT devices are usually integrated into the information network that allows them to communicate and exchange data with other devices and systems.

1.3 Why IoT?

When something is connected to the internet, that means that it can send information or receive information, or both. This ability to send and/or receive information makes things “smart.”

Let's use smartphones again as an example. Right now you can listen to just about any song in the world, but it's not because your phone actually has every song in the world stored on it. It's because every song in the world is stored somewhere else, but

your phone can send information (asking for that song) and then receive information (streaming that song on your phone).

To be smart, a thing doesn't need to have super storage or a super computer inside of it - it just needs access to it. All a thing has to do is connect to super storage or to a super computer. In the Internet of Things, all the things that are being connected to the internet can be put into three categories:

Things that collect information and then send it.

Things that receive information and then act on it.

Things that do both.

And all three of these have enormous benefits that compound on each other.

1. Collecting and Sending Information

Sensors could be temperature sensors, motion sensors, moisture sensors, air quality sensors, light sensors, you name it. These sensors, along with a connection, allow us to automatically collect information from the environment which, in turn, allows us to make more intelligent decisions.

On a farm, automatically getting information about the soil moisture can tell farmers exactly when their crops need to be watered. Instead of watering too much (which can be an expensive over-use of irrigation systems) or watering too little (which can be an expensive loss of crops), the farmer can ensure that crops get exactly the right amount of water. This enables farmers to increase their crop yield while decreasing their associated expenses.

Just as our sight, hearing, smell, touch, and taste allow us, humans, to make sense of the world, sensors allow machines (and the humans monitoring the machines) to make sense of the world.

2. Receiving and Acting on Information

We're all very familiar with machines getting information and then acting. Your printer receives a document and it prints it. Your car receives a signal from your car keys and the doors open. The examples are endless.

Whether it's a simple as sending the command "turn on" or as complex as sending a 3D model to a 3D printer, we know that we can tell machines what to do from far away. So what?

The real power of the Internet of Things arises when things can do both of the above. Things that collect information and send it, but also receive information and act on it.

3. Doing Both: The Goal of an IoT System

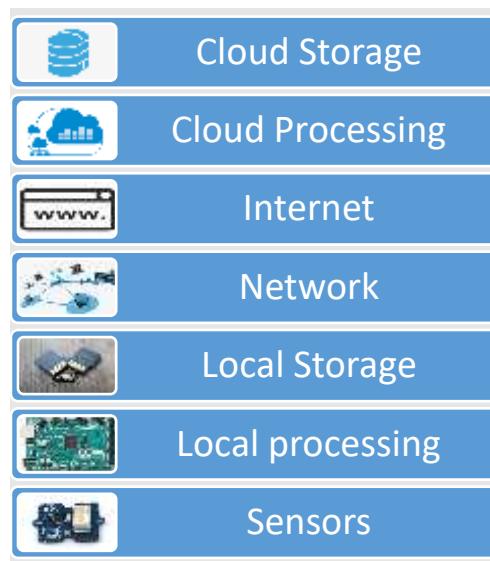
Let's quickly go back to the farming example. The sensors can collect information about the soil moisture to tell the farmer how much to water the crops, but you don't actually need the farmer. Instead, the irrigation system can automatically turn on as needed, based on how much moisture is in the soil.

You can take it a step further too. If the irrigation system receives information about the weather from its internet connection, it can also know when it's going to rain and decide not to water the crops today because they'll be watered by the rain anyways.

And it doesn't stop there! All this information about the soil moisture, how much the irrigation system is watering the crops, and how well the crops actually grow can be collected and sent to supercomputers that run amazing algorithms that can make sense of all this information.

And that's just one kind of sensor. Add in other sensors like light, air quality, and temperature, and these algorithms can learn much, much more. With dozens, hundreds, thousands of farms all collecting this information, these algorithms can create incredible insights into how to make crops grow the best, helping to feed the world.

1.4 IoT Architecture



Sensors

They transform analog data given from scanning the environment to digital data, but they merely do any processing. On the bright side, they don't consume much power

and can live on batteries for a long time. Sensors are present in everyday life more than you would expect. They improve industry, agriculture, homes, transportation or smart phones for example. They are tools which help monitoring the environment, collecting data about it and, with the help of computers, acting accordingly.

Local processing and storage devices

Local processing devices are the second level and third in IoT. At this point, data is locally stored and processed, ideally not sent forwards unless relevant. This part is explained in detail in the hardware section, as said devices are nothing more than microcontrollers and embeded boards, which handle the data they receive from the sensors.

Network and Internet

There is hardware which connects to the previously described devices, pulls out data and sends it to the cloud to be stored. There are 4 protocols used at this level: CoAP, MQTT (less secure and designed for machine to machine communication), HTTP (web protocol) and XMPP which functions as a chat.

Cloud

In the cloud, which comes next, data is collected and the main goal is for it to reach the point of making predictions based on the stored information. The cloud however, even though it represents one of the most useful features of the internet, is not used properly. Data sent to the cloud didn't reach the level of being formerly processed. Which means there is no preselected data. The cloud is constantly loaded with irrelevant information and thus losing it's property of being practical.

1.5 What is industrial IoT?

Industrial IoT (IIoT) refers to the application of IoT technology in industrial settings, especially with respect to instrumentation and control of sensors and devices that engage cloud technologies. Recently, industries have used machine-to-machine communication (M2M) to achieve wireless automation and control. But with the emergence of cloud and allied technologies (such as analytics and machine learning), industries can achieve a new automation layer and with it create new revenue and business models. IIoT is sometimes called the fourth wave of the industrial revolution, or Industry 4.0. The following are some common uses for IIoT:

- Smart manufacturing
- Connected assets and preventive and predictive maintenance
- Smart power grids
- Smart cities

- Connected logistics
- Smart digital supply chains

1.6 IoT Applications by Industries

The ubiquity of the Internet of Things is a fact of life thanks to its adoption by a wide range of industries. IoT's versatility makes it an attractive option for so many businesses, organizations, and government branches, that it doesn't make sense to ignore it. Let us learn about IoT applications across industries below:

IoT Applications in Agriculture

For indoor planting, IoT makes monitoring and management of micro-climate conditions a reality, which in turn increases production. For outside planting, devices using IoT technology can sense soil moisture and nutrients, in conjunction with weather data, better control smart irrigation and fertilizer systems. If the sprinkler systems dispense water only when needed, for example, this prevents wasting a precious resource.

IoT Applications in Consumer Use

For the private citizen, IoT devices in the form of wearables and smart homes make life easier. Wearables cover accessories such as Fitbit, smartphones, Apple watches, health monitors, to name a few. These devices improve entertainment, network connectivity, health, and fitness.

Smart homes take care of things like activating environmental controls so that your house is at peak comfort when you come home. Dinner that requires either an oven or a crockpot can be started remotely, so the food is ready when you arrive. Security is made more accessible as well, with the consumer having the ability to control appliances and lights remotely, as well as activating a smart lock to allow the appropriate people to enter the house even if they don't have a key.

IoT Applications in Healthcare

First and foremost, wearable IoT devices let hospitals monitor their patients' health at home, thereby reducing hospital stays while still providing up to the minute real-time information that could save lives. In hospitals, smart beds keep the staff informed as to the availability, thereby cutting wait time for free space. Putting IoT sensors on critical equipment means fewer breakdowns and increased reliability, which can mean the difference between life and death.

Elderly care becomes significantly more comfortable with IoT. In addition to the above-mentioned real-time home monitoring, sensors can also determine if a patient has fallen or is suffering a heart attack.

IoT Applications in Insurance

Even the insurance industry can benefit from the IoT revolution. Insurance companies can offer their policyholders discounts for IoT wearables such as Fitbit. By employing fitness tracking, the insurer can offer customized policies and encourage healthier habits, which in the long run, benefits everyone, insurer, and customer alike.

IoT Applications in Manufacturing

The world of manufacturing and industrial automation is another big winner in the IoT sweepstakes. RFID and GPS technology can help a manufacturer track a product from its start on the factory floor to its placement in the destination store, the whole supply chain from start to finish. These sensors can gather information on travel time, product condition, and environmental conditions that the product was subjected to.

Sensors attached to factory equipment can help identify bottlenecks in the production line, thereby reducing lost time and waste. Other sensors mounted on those same machines can also track the performance of the machine, predicting when the unit will require maintenance, thereby preventing costly breakdowns.

IoT Applications in Retail

IoT technology has a lot to offer the world of retail. Online and in-store shopping sales figures can control warehouse automation and robotics, information gleaned from IoT sensors. Much of this relies on RFIDs, which are already in heavy use worldwide.

Mall locations are iffy things; business tends to fluctuate, and the advent of online shopping has driven down the demand for brick and mortar establishments. However, IoT can help analyze mall traffic so that stores located in malls can make the necessary adjustments that enhance the customer's shopping experience while reducing overhead.

Speaking of customer engagement, IoT helps retailers target customers based on past purchases. Equipped with the information provided through IoT, a retailer could craft a personalized promotion for their loyal customers, thereby eliminating the need for costly mass-marketing promotions that don't stand as much of a chance of success. Much of these promotions can be conducted through the customers' smartphones, especially if they have an app for the appropriate store.

IoT Applications in Transportation

By this time, most people have heard about the progress being made with self-driving cars. But that's just one bit of the vast potential in the field of transportation. The GPS, which, if you think of it, is another example of IoT, is being utilized to help transportation

companies plot faster and more efficient routes for trucks hauling freight, thereby speeding up delivery times.

There's already significant progress made in navigation, once again alluding to a phone or car's GPS. But city planners can also use that data to help determine traffic patterns, parking space demand, and road construction and maintenance.

There's even a possibility that apps can be made that can prevent a car from starting if the driver is inebriated!

IoT Applications in Utilities/Energy

IoT sensors can be employed to monitor environmental conditions such as humidity, temperature, and lighting. The information provided by IoT sensors can aid in the creation of algorithms that regulate energy usage and make the appropriate adjustments, eliminating the human equation (and let's face it, who of us hasn't forgotten to switch off lights in a room or turn down the thermostat?).

With IoT-driven environmental control, businesses and private residences can experience significant energy savings, which in the long run, benefits everyone, including the environment!

On a larger scale, data gathered by the Internet of Things can be used to help run municipal power grids more efficiently, analyzing factors such as usage. Also, the sensors can help pinpoint outages faster, thereby increasing the response time of repair crews and decreasing blackout times.

1.7 The Future of the Internet of Things

So, considering the above, just what does the future have in store for the Internet of Things?

A Gartner report predicts that connected devices across all manner of technologies will hit 20.6 billion. That number could hit 1 trillion by 2025, according to HP, and that's just a staggering figure. According to a Cisco report, the next decade will see IoT devices creating \$14.4 trillion worth of value across several industries like the ones mentioned above.

In other words, the Internet of Things is poised to create life-changing conditions in our lives, both in a professional and personal capacity. Many of the innovations mentioned are already in place to one extent or another. One thing's for sure: there's no going back. The IoT offers an unprecedented degree of control and efficiency that no industry can ignore.

Unit 2: Sensors and Actuators

Learning Outcomes:

- Understand the Concept of Sensors and Actuators
- Able to identify various sensors and actuators
- Understand the use and applications of Various Sensors and Actuators

2.1 Sensors

The era of automation has begun already. Most of the things that we use now can be automated. To design automated devices first we need to know about the sensors, these are the modules/devices which are helpful in making things done without human intervention. Even the mobiles or smartphones which we daily use will have some sensors like hall sensor, proximity sensor, accelerometer, touch screen, microphone etc. These sensor acts as eyes, ears, nose of any electrical equipment which senses the parameters in outside world and give readings to devices or Microcontroller.

2.1.1 What is Sensor?

The sensor can be defined as a device which can be used to sense/detect the physical quantity like force, pressure, strain, light etc and then convert it into desired output like the electrical signal to measure the applied physical quantity. In few cases, a sensor alone may not be sufficient to analyze the obtained signal. In those cases, a **signal conditioning unit** is used in order to maintain sensor's output voltage levels in the desired range with respect to the end device that we use.

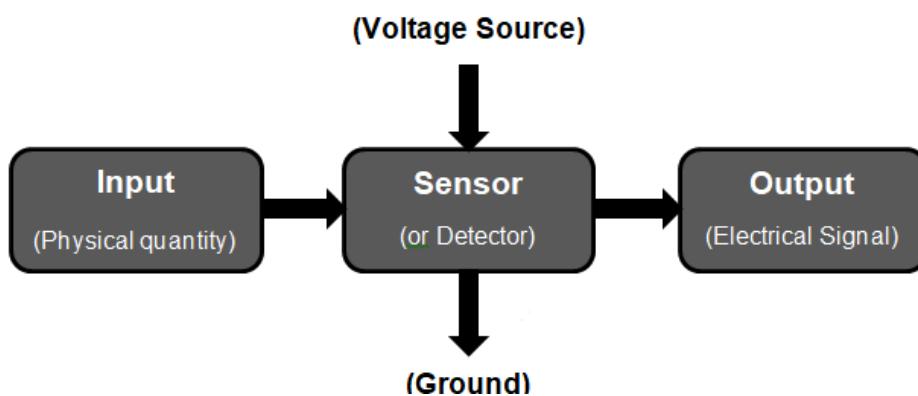


Fig:Block Diagram of Sensor

In **signal conditioning unit**, the output of the sensor may be amplified, filtered or modified to the desired output voltage. For example, if we consider a microphone, it detects the audio signal and converts to the output voltage (is in terms of millivolts) which becomes hard to drive an output circuit. So, a signal conditioning unit (an amplifier) is used to increase the signal strength. But the signal conditioning may not be necessary for all the sensors like photodiode, LDR etc.

Most of the sensors can't work independently. So, sufficient input voltage should be applied to it. Various sensors have different operating ranges which should be considered while working with it else the sensor may get damaged permanently.

2.1.2 Characteristics of Sensors

A good sensor should have the following characteristics

- High Sensitivity: Sensitivity indicates how much the output of the device changes with unit change in input (quantity to be measured). For example, the voltage of a temperature sensor changes by 1mV for every 1°C change in temperature than the sensitivity of the sensor is said to be 1mV/°C.
- Linearity: The output should change linearly with the input.
- High Resolution: Resolution is the smallest change in the input that the device can detect.
- Less Noise and Disturbance.
- Less power consumption.

Sensors and Electronic Concepts

A sensor is a device that measures physical input from its environment and converts it into data that can be interpreted by either a human or a machine. Most sensors are electronic (the data is converted into electronic data).

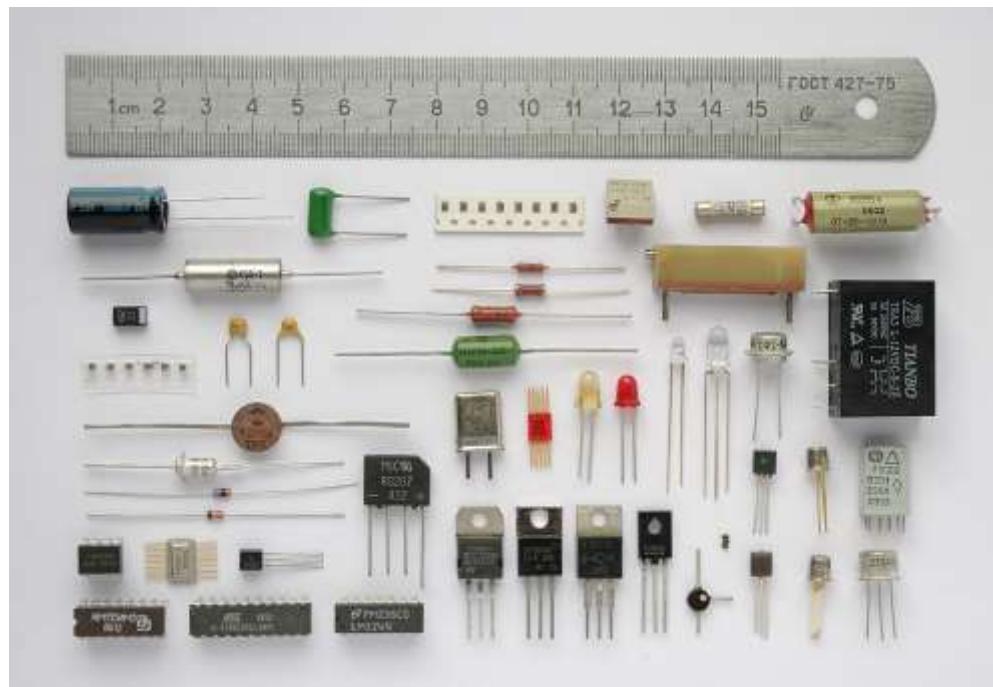
Let's look at Electronic Concepts.



Reference: <https://thumbs.dreamstime.com/z/let-s-start-29574630.jpg>

2.2 Electronics Components

An electronic component is a physical entity in an electronic system used to affect movement of electrons. Electronic components have a number of electronic terminals or leads which connect to other electronic components over wire to create an electronic circuit.

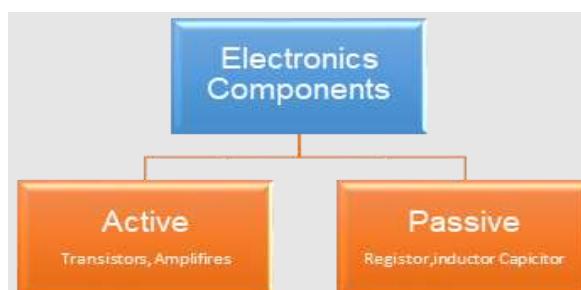


Reference: <https://en.wikipedia.org/wiki/File:Componentes.JPG>

Classification of Electronics Components

They can be classified into two types i.e., Active Components and Passive Components. Active elements are those which possess gain. They can give energy to the circuit. On the contrary passive elements do not possess gain and they cannot give energy continuously to the circuit.

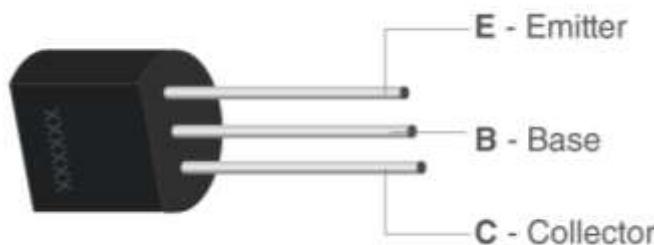
Passive components cannot amplify or energize the energy of the signal associated with them, they can only attenuate it, while active components can energize or amplify the signal.



Transistor

A transistor is a semiconductor device used to amplify and switch electronic signals and electrical power.

Transistor is formed from two words “Transfer” and “Resistor”. Thus, it transfers resistance from one part of circuit to other. If at input side the resistance is high then the resistance at the output side will be low. Transistor is a three terminal device which can act as a switch or an amplifier. It can be controlled either by voltage or current thus it is called voltage-controlled device or a current controlled device.



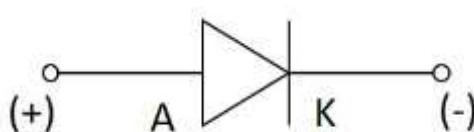
Reference: <https://cdn1.byjus.com/wp-content/uploads/2020/02/Transistor-1.png>

Transistor applications

- Transistor have application in both Analog and Digital Circuit
- Power Regulator Circuits
- Microprocessor ICs
- Mobile Phone charger
- AC to DC adaptors
- TV Power supply section
- SMPS
- Electronics Switching devices etc.

Diode

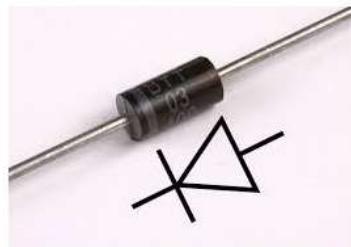
A semiconductor diode is a two terminal electronic component with a PN junction. This is also called as a Rectifier.



Symbol of a Diode

The anode which is the positive terminal of a diode is represented with A and the cathode, which is the negative terminal is represented with K. To know the anode

and cathode of a practical diode, a fine line is drawn on the diode which means cathode, while the other end represents anode.



Representing anode and cathode of a practical diode through its symbol

Biassing of a Diode

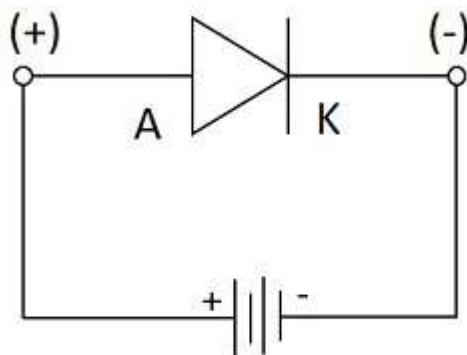
When a diode or any two-terminal component is connected in a circuit, it has two biased conditions with the given supply. They are Forward biased condition and Reverse biased condition.

Forward Biased Condition

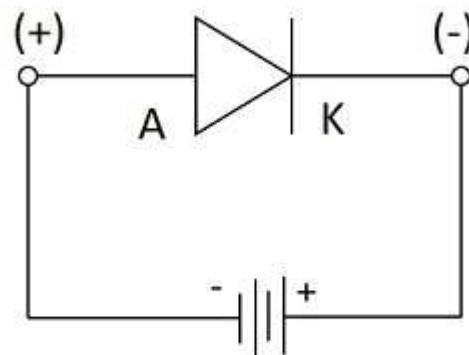
When a diode is connected in a circuit, with its anode to the positive terminal and cathode to the negative terminal of the supply, then such a connection is said to be forward biased condition.

Reverse Biased Condition

When a diode is connected in a circuit, with its anode to the negative terminal and cathode to the positive terminal of the supply, then such a connection is said to be Reverse biased condition.



Forward biased Connection

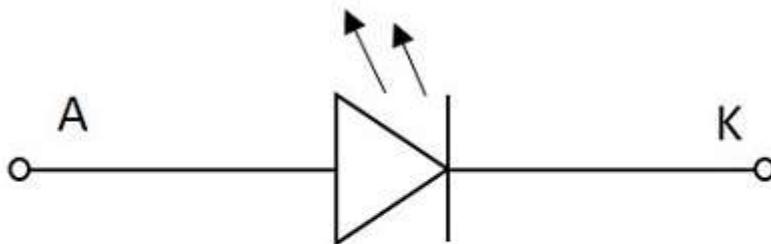


Reverse biased Connection

LED – Light Emitting Diodes

This one is the most popular diodes used in our daily life. This is also a normal PN junction diode except that instead of silicon and germanium, the materials like gallium arsenide, gallium arsenide phosphide are used in its construction.

The figure below shows the symbol of a Light emitting diode.



Symbol of LED

Like a normal PN junction diode, this is connected in forward bias condition so that the diode conducts. The conduction takes place in a LED when the free electrons in the conduction band combine with the holes in the valence band. This process of recombination **emits light**. This process is called as **Electroluminescence**. The color of the light emitted depends upon the gap between the energy bands.

The LEDs for non-visible Infrared light are used mostly in remote controls.



Reference: https://www.tutorialspoint.com/basic_electronics/basic_electronics_optoelectric_diodes.htm

Applications of LED

There are many applications for LED such as –

In Displays

- Especially used for seven segment display
- Digital clocks
- Microwave ovens
- Traffic signalling
- Display boards in railways and public places
- Toys

In Electronic Appliances

- Stereo tuners
- Calculators
- DC power supplies
- On/Off indicators in amplifiers
- Power indicators

Commercial Use

- Infrared readable machines
- Barcode readers
- Solid state video displays

Optical Communications

- In Optical switching applications
- For Optical coupling where manual help is unavailable
- Information transfer through FOC
- Image sensing circuits
- Burglar alarms
- In Railway signalling techniques
- Door and other security control systems

Resistor

Resistor is an electrical component that reduces the electric current. The resistor's ability to reduce the current is called resistance and is measured in units of ohms (symbol: Ω). The resistor's resistance limits the flow of electrons through a circuit. Resistors come in a variety of shapes and sizes.

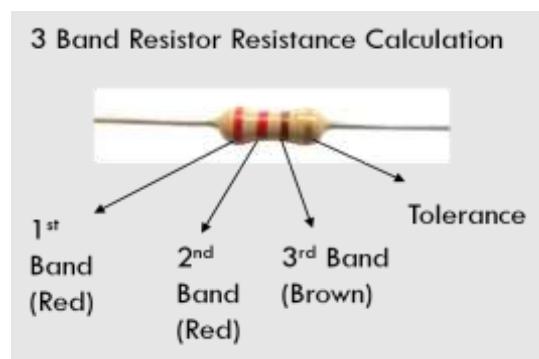
Resistor Color Code Chart

The following color code chart is used to calculate the value of resistance of a resistor. All Resistors follow a colour coded pattern as show below.

Color	Color	1st Band	2nd Band	3rd Band Multiplier	4th Band Tolerance
Black		0	0	x1Ω	
Brown		1	1	x10Ω	±1%
Red		2	2	x100Ω	±2%
Orange		3	3	x1kΩ	
Yellow		4	4	x10kΩ	
Green		5	5	x100kΩ	±0.5%
Blue		6	6	x1MΩ	±0.25%
Violet		7	7	x10MΩ	±0.10%
Grey		8	8	x100MΩ	±0.05%
White		9	9	x1GΩ	
Gold				x0.1Ω	±5%
Silver				x0.01Ω	±10%

Reference: <https://sites.google.com/view/trainingday-1/basics-of-electronics/components?authuser=0>

How to Calculate Resistance of a Resistor?



Reference: <https://sites.google.com/view/trainingday-1/basics-of-electronics/components?authuser=0>

Resistance = 1st band value (first digit) 2nd band value (second digit) multiplied by multiplier value of 3rd band.

$$= 1^{\text{st}} \text{ band } 2^{\text{nd}} \text{ band} \times 3^{\text{rd}} \text{ band} \text{ (Multiplier)}$$

$$= 22 \times 10 \Omega(\text{ohms})$$

$$= 220 \Omega \text{ (ohms)}$$

Ohm's Law

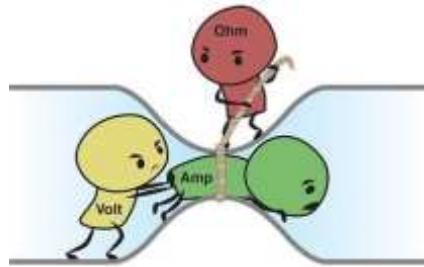
Ohm's law states that the current through a conductor between two points is directly proportional to the voltage or potential difference between the two points provided the temperature is constant for a constant length and area.

Ohm's Law Formula

$$\text{Voltage} = \text{Current} \times \text{Resistance}$$

$$V = I \times R$$

Where, V= voltage (Unit: volts or V), I= current (Unit: Amperes or A) and R= resistance (Unit: ohms or Ω)



Reference: <https://sites.google.com/view/trainingday-1/basics-of-electronics/components?authuser=0>

Buzzer



Reference: <https://www.instructables.com/How-to-use-a-Buzzer-Arduino-Tutorial/>

There are many ways to communicate between the user and a product. One of the best ways is audio communication using a buzzer

What is a Buzzer?

An audio signalling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers,

alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

The **pin configuration of the buzzer** is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-' symbol or short terminal and it is connected to the GND terminal.



Reference: <https://www.elprocus.com/wp-content/uploads/Buzzer-Pin-Configuration-272x300.jpg>

How to use a Buzzer?

A buzzer is an efficient component to include the features of sound in our system or project. It is an extremely small & solid two-pin device thus it can be simply utilized on breadboard or PCB. So, in most applications, this component is widely used.

There are two kinds of buzzers commonly available like simple and readymade. Once a simple type is power-driven then it will generate a beep sound continuously. A readymade type looks heavier & generates a Beep. Beep. Beep. This sound is because of the internal oscillating circuit within it.

This buzzer uses a DC power supply that ranges from 4V – 9V. To operate this, a 9V battery is used but it is suggested to utilize a regulated +5V/+6V DC supply. Generally, it is connected through a switching circuit to switch ON/OFF the buzzer at the necessary time interval.

2.3 Electronics Signal

A Signal can be understood as "a representation that gives some information about the data present at the source from which it is produced." This is usually time varying. Hence, a signal can be a source of energy which transmits some information. This can easily be represented on a graph.

A signal can be of any type that conveys some information. This signal produced from an electronic equipment, is called as Electronic Signal or Electrical Signal. These are generally time variants.

Types of Signals

Signals can be classified either as Analog or Digital, depending upon their characteristics.

- Analog Signal
- Digital Signal

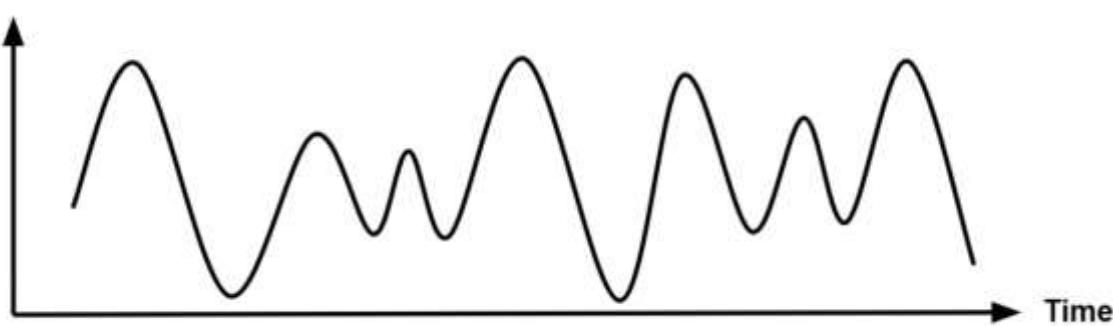
Analog Signal

An analog signal is time-varying and generally bound to a range (e.g. +12V to -12V), but there is an infinite number of values within that continuous range. An analog signal uses a given property of the medium to convey the signal's information, such as electricity moving through a wire. In an electrical signal, the voltage, current, or frequency of the signal may be varied to represent the information.

Analog signals are often calculated responses to changes in light, sound, temperature, position, pressure, or other physical phenomena.

When plotted on a voltage vs. time graph, an analog signal should produce a smooth and continuous curve. There should not be any discrete value changes

Amplitude



Reference: <https://www.monolithicpower.com/en/analog-vs-digital-signal>

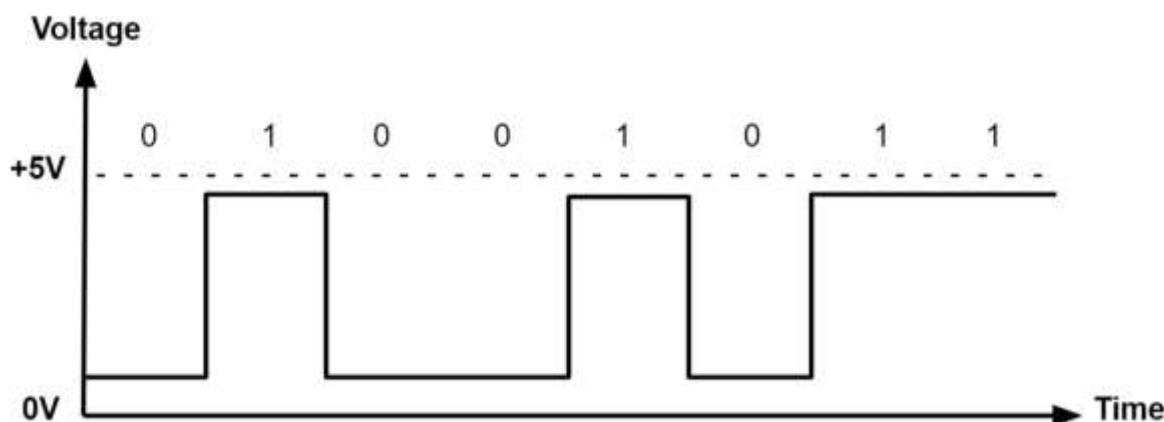
Digital Signal

A digital signal is a signal that represents data as a sequence of discrete values. A digital signal can only take on one value from a finite set of possible values at a given

time. With digital signals, the physical quantity representing the information can be many things:

1. Variable electric current or voltage
2. Phase or polarization of an electromagnetic field
3. Acoustic pressure
4. The magnetization of a magnetic storage media

Digital signals are used in all digital electronics, including computing equipment and data transmission devices. When plotted on a voltage vs. time graph, digital signals are one of two values, and are usually between 0V and VCC (usually 1.8V, 3.3V, or 5V)



Reference: <https://www.monolithicpower.com/en/analog-vs-digital-signal>

2.4 PWM – Pulse Width Modulation

In power electronics, pulse width modulation is a proven effective technique that is used to control semiconductor devices. Pulse width modulation or PWM is a commonly used control technique that generates analog signals from digital devices such as microcontrollers. The signal thus produced will have a train of pulses, and these pulses will be in the form of square waves.

What is Pulse Width Modulation?

Pulse width modulation reduces the average power delivered by an electrical signal by converting the signal into discrete parts. In the PWM technique, the signal's energy

is distributed through a series of pulses rather than a continuously varying (analog) signal.

Important Parameters associated with PMW signal

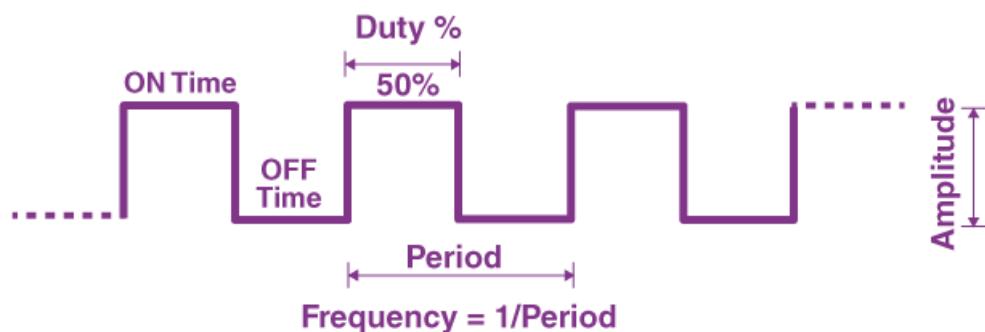
Duty Cycle of PMW

As we know, a PWM signal stays “ON” for a given time and stays “OFF” for a certain time. The percentage of time for which the signal remains “ON” is known as the duty cycle. If the signal is always “ON,” then the signal must have a 100 % duty cycle. The formula to calculate the duty cycle is given as follows:

$$\text{Duty Cycle} = \frac{\text{Turn ON time}}{\text{Turn ON time} + \text{Turn OFF time}}$$

The average value of the voltage depends on the duty cycle. As a result, the average value can be varied by controlling the width of the “ON” of a pulse.

Frequency of PMW



Reference: <https://cdn1.byjus.com/wp-content/uploads/2021/01/duty-cycle-of-pulse-width-modulation.png>

The frequency of PMW determines how fast a PMW completes a period. The frequency of a pulse is shown in the figure above.

The frequency of PMW can be calculated as follows:

$$\text{Frequency} = 1/\text{Time Period}$$

$$\text{Time Period} = \text{ON time} + \text{OFF time}$$

Output Voltage of PWM signal

The output voltage of the PWM signal will be the percentage of the duty cycle. For example, for a 100% duty cycle, if the operating voltage is 5 V then the output voltage will also be 5 V. If the duty cycle is 50%, then the output voltage will be 2.5v.

Applications of Pulse Width Modulation

Due to the high efficiency, low power loss, and the PWM technique's ability to precisely control the power, the technique is used in a variety of power applications. Some of the applications of PWM are as follows:

- The pulse width modulation technique is used in telecommunication for encoding purposes.
- The PWM helps in voltage regulation and therefore is used to control the speed of motors.
- The PWM technique controls the fan inside a CPU of the computer, thereby successfully dissipating the heat.
- PWM is used in Audio/Video Amplifiers.

2.5 ADC – Analog to Digital Converter

Analog-to-Digital converters (ADC) translate analog signals, real world signals like temperature, pressure, voltage, current, distance, or light intensity, into a digital representation of that signal. This digital representation can then be processed, manipulated, computed, transmitted or stored.

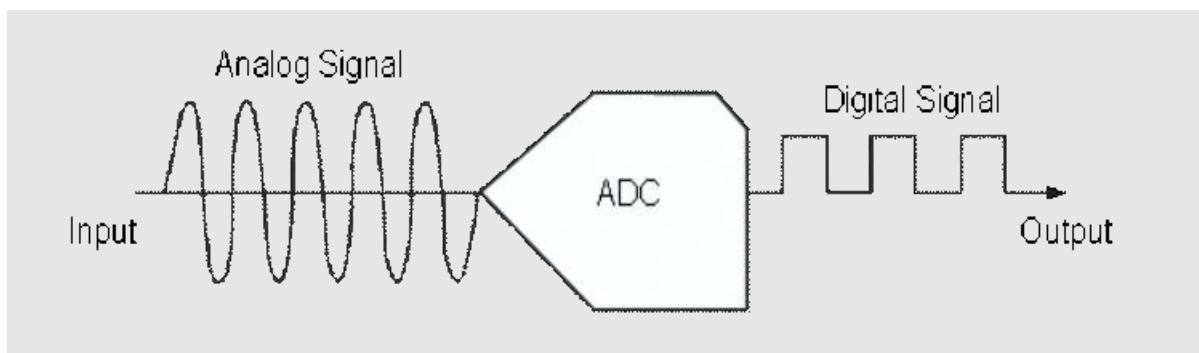


Fig: Analog to Digital Conversion

Reference: <https://wiki.analog.com/university/courses/electronics/text/chapter-20>

Digital signals are represented by a sequence of discrete values where the signal is broken down into sequences that depend on the time series or sampling rate. The easiest way to explain this is through a visual! Figure shows a great example of what analog and digital signals look like.

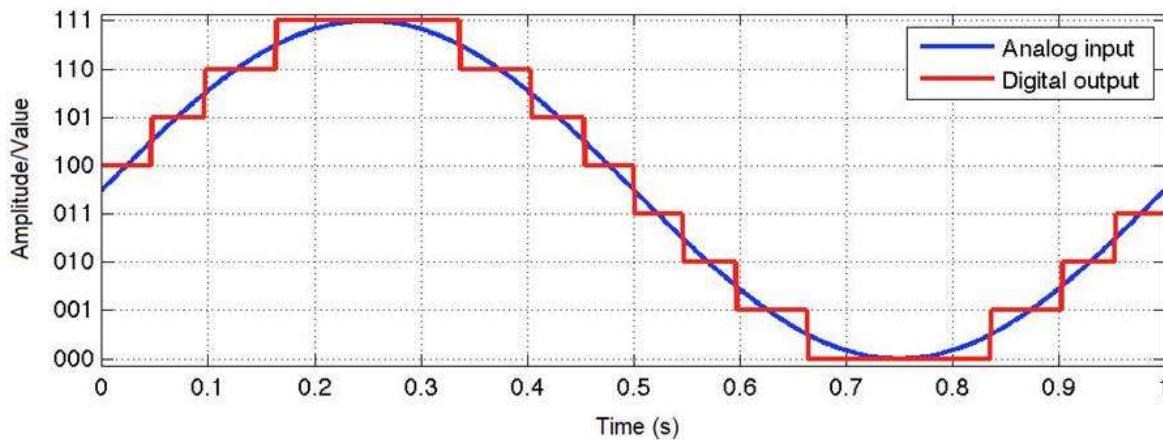


Fig: continuous signal (analog) turning into a digital signal.

Reference: <https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters#:~:text=ADCs%20follow%20a%20sequence%20when,its%20sampling%20rate%20and%20resolution.>

Microcontrollers can't read values unless it's digital data. This is because microcontrollers can only see "levels" of the voltage, which depends on the resolution of the ADC and the system voltage.

ADCs follow a sequence when converting analog signals to digital. They first sample the signal, then quantify it to determine the resolution of the signal, and finally set binary values and send it to the system to read the digital signal. Two important aspects of the ADC are its sampling rate and resolution.

Applications of Analog to Digital Converter

The applications of ADC include the following.

1. AC (air conditioner) includes temperature sensors to maintain the temperature within the room. So this conversion of temperature can be done from analog to digital with the help of ADC.
2. It is also used in a digital oscilloscope to convert the signal from analog to digital to display.
3. ADC is used to convert the analog voice signal to digital in mobile phones because mobile phones use digital voice signals but actually, the voice signal is in the form of analog. So ADC is used to convert the signal before sending the signal toward the transmitter of the cell phone.
4. ADC is used in medical devices like MRI and X-Ray to convert the images from analog to digital before alteration.
5. The camera in the mobile mainly used for capturing images as well as videos. These are stored in the digital device, so these are converted to digital form using ADC.

6. The cassette music can also be changed into a digital like CDS & thumb drives use ADC.
7. At present ADC is used in every device because almost all devices available in the market are in digital version. So these devices use ADC.

Types sensors

Sensors are split up into four main categories. Such as,

- Analog Sensor
- Digital Sensor
- Active Sensor
- Passive Sensor

Analog Sensors

The sensor that produces continuous signal with respect to time with analog output is called as Analog sensors. The analog output generated is proportional to the measured or the input given to the system. Generally, analog voltage in the range of 0 to 5 V or current is produced as the output. The various physical parameters like temperature, stress, pressure, displacement, etc. are examples for continuous signals. Examples: accelerometers, speed sensors, pressure sensors, light sensors, temperature sensors.

Digital Sensors

When data is converted and transmitted digitally, it is called as Digital sensors. Digital sensors are the one, which produces discrete output signals. Discrete signals will be non-continuous with time and it can be represented in "bits" for serial transmission and in "bytes" for parallel transmission. The measuring quantity will be represented in digital format. Digital output can be in form of Logic 1 or logic 0 (ON or OFF). A digital sensor consists of sensor, cable and a transmitter. The measured signal is converted into a digital signal inside sensor itself without any external component. Cable is used for long distance transmission.

Active Sensor

Based on power requirement sensors can be classified as active and passive

Active sensors are those which do not require external power source for their functioning. They generate power within themselves to operate and hence called as self-generating type. The energy for functioning is derived from the quantity being measured. For example, piezoelectric crystal generates electrical output (charge) when subjected to acceleration.

Passive sensors

Passive sensors require external power source for their functioning. Most of the resistive, inductive and capacitive sensors are passive (just as resistors, inductors and capacitors are called passive devices).

Applications of Sensors

1. Automobile
2. Manufacturing
3. Agriculture
4. Aviation
5. Medical & Healthcare sector etc.

Let's discuss different types of sensors

2.6 Types of sensors

2.6.1 Rotary Angle Sensor or Potentiometer

The rotary angle sensor produces analog output between 0 and Vcc (5V DC with Seeeduino) on its D1 connector. The D2 connector is not used. The angular range is 300 degrees with a linear change in value. The resistance value is 10k ohms, perfect for Arduino use. This may also be known as a "potentiometer".



Reference: <https://robu.in/wp-content/uploads/2019/12/Grove-Rotary-Angle-Sensor.jpg>

2.6.2 Sound Sensor

1. The sound sensor is one type of module used to notice the sound. Generally, this module is used to detect the intensity of sound. The applications of this module mainly include switch, security, as well as monitoring. The accuracy of this sensor can be changed for the ease of usage.

2. This sensor employs a microphone to provide input to buffer, peak detector and an amplifier. This sensor notices a sound, & processes an o/p voltage signal to a microcontroller. After that, it executes required processing.
3. This sensor is capable to determine noise levels within DB's or decibels at 3 kHz to 6 kHz frequencies approximately wherever the human ear is sensitive. In smartphones, there is an android application namely decibel meter used to measure the sound level.

Sound Sensor Pin Configuration

This sensor includes three pins which include the following.



Reference: https://files.seeedstudio.com/wiki/Grove_Sound_Sensor/img/page_small_1.jpg

- Pin1 (VCC): 3.3V DC to 5V DC
- Pin2 (GND): This is a ground pin
- Pin3 (DO): This is an output pin

Working Principle

The working principle of this sensor is related to human ears. Because human ear includes a diaphragm and the main function of this diaphragm is, it uses the vibrations and changes into signals. Whereas in this sensor, it uses a microphone and the main function of this is, it uses the vibrations and changes into current otherwise voltage. Generally, it includes a diaphragm which is designed with magnets that are twisted with metal wire. When sound signals hit the diaphragm, then magnets within the sensor vibrates & simultaneously current can be stimulated from the coils.

Features

The features of the sound sensor include the following

- These sensors are very simple to use
- It gives analog o/p signal
- Simply incorporates using logic modules on the input area

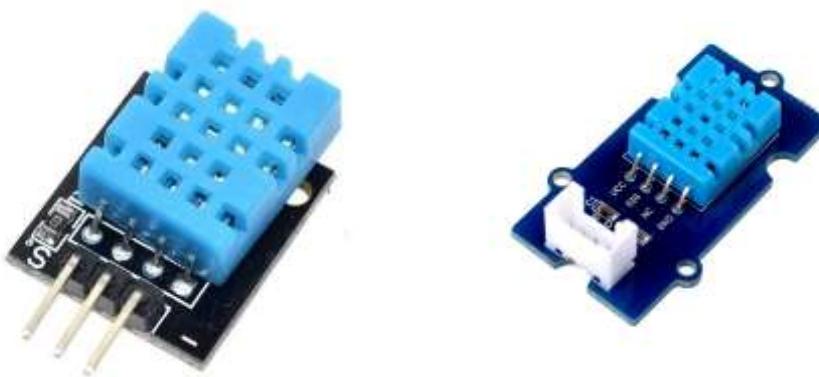
Application

- Security system for Office or Home
- Spy Circuit
- Home Automation
- Robotics
- Smart Phones
- Ambient sound recognition
- Audio amplifier
- Sound level recognition (not capable to obtain precise dB value)

2.6.3 Temperature and humidity sensor

Temperature and humidity sensor (or rh temp sensor) is devices that can convert temperature and humidity into electrical signals that can easily measure temperature and humidity. Temperature humidity transmitters on the market generally measure the amount of temperature and relative humidity in the air, and convert it into electrical signals or other signal forms according to certain rules and output the device to the instrument or software to meet the environmental monitoring needs of users.

1. DHT11 is a Humidity and Temperature Sensor, which generates calibrated digital output. DHT11 can be interface with any microcontroller like Arduino, Raspberry Pi, etc. and get instantaneous results. DHT11 is a low-cost humidity and temperature sensor which provides high reliability and long-term stability.
2. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and outputs a digital signal on the data pin (no analog input pins needed). It's very simple to use, and libraries and sample codes are available for Arduino and Raspberry Pi.
3. This module makes it easy to connect the DHT11 sensor to an Arduino or microcontroller as it includes the pull up resistor required to use the sensor. Only three connections are required to be made to use the sensor - Vcc, Gnd and Output.



Reference: https://files.seeedstudio.com/wiki/Grove-TemperatureAndHumidity_Sensor/img/list.jpg

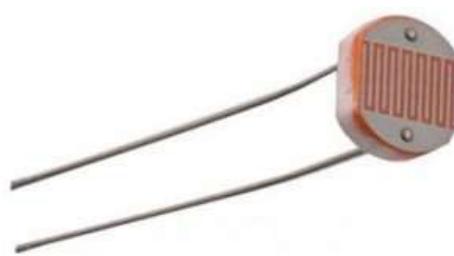
4. It has high reliability and excellent long-term stability, thanks to the exclusive digital signal acquisition technique and temperature & humidity sensing technology.

2.6.4 Light Sensor

The light sensor is used to discover the light & generates a voltage difference. The light sensors used in robots are two types photovoltaic cells & photoresistors. Photovoltaic cells are used to change the solar radiation energy to electrical and these sensors are used in solar robot manufacturing. Photoresistors are used to alter their resistance by modifying light intensities. When light is more on it then resistance will be less. These light sensors are not expensive, so used easily in robots.

LDR is one the sensor belongs to light sensor family. An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.

As its name implies, the **Light Dependent Resistor** (LDR) is made from a piece of exposed semiconductor material such as cadmium sulphide that changes its electrical resistance from several thousand Ohms in the dark to only a few hundred Ohms when light falls upon it by creating hole-electron pairs in the material.



Light Sensor



Light Sensor

The most common type of LDR has a resistance that falls with an increase in the light intensity falling upon the device (as shown in the image above). The resistance of an LDR may typically have the following resistances:

Daylight = 5000Ω

Dark = 20000000Ω

You can therefore see that there is a large variation between these figures. If you plotted this variation on a graph, you would get something similar to that shown by the graph shown above.

Applications of LDRs

There are many applications for Light Dependent Resistors. These include:

- **Lighting switch**

The most obvious application for an LDR is to automatically turn on a light at a certain light level. An example of this could be a street light or a garden light.

- **Camera shutter control**

LDRs can be used to control the shutter speed on a camera. The LDR would be used to measure the light intensity which then adjusts the camera shutter speed to the appropriate level.

2.6.5 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e., the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

Ultrasonic sensors work by emitting sound waves at a frequency which is too high for humans to hear.



Fig: Ultrasonic Sensor

Reference: <https://robu.in/ultrasonic-sensor-working-principle/>

An above image shows the HC-SR-04 ultrasonic sensor which has a transmitter, receiver. The pin configuration is,

1. VCC - +5 V supply
2. TRIG – Trigger input of the sensor. Microcontroller applies 10 us trigger pulse to the HC-SR04 ultrasonic module.
3. ECHO–Echo output of the sensor. Microcontroller reads/monitors this pin to detect the obstacle or to find the distance.
4. **GND** – Ground

Sound is a mechanical wave travelling through the mediums, which may be a solid, or liquid or gas. Sound waves can travel through the mediums with specific velocity depends on the medium of propagation. The sound waves which are having high frequency reflect from boundaries and produce distinctive echo patterns.

Features of an Ultrasonic Sensor

1. Supply voltage: 5V (DC).
2. Supply current: 15mA.
3. Modulation frequency: 40Hz.
4. Output: 0 – 5V (Output high when obstacle detected in range).
5. Beam Angle: Max 15 degrees.
6. Distance: 2 cm – 400 cm.
7. Accuracy: 0.3cm.
8. Communication: Positive TTL pulse.

Ultrasonic Sensor Working Principle

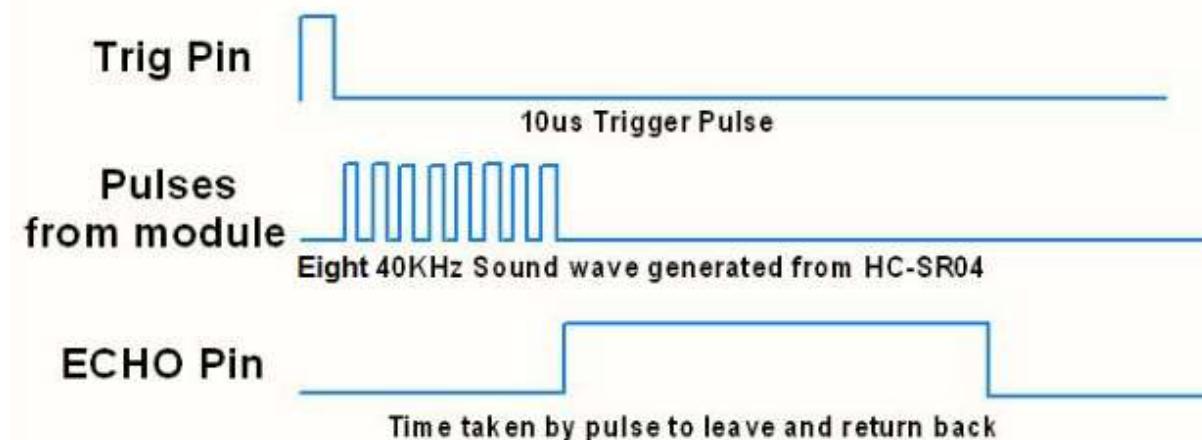
Ultrasonic sensors emit short, high-frequency sound pulses at regular intervals. These propagate in the air at the velocity of sound. If they strike an object, then they reflected back as an echo signal to the sensor, which itself computes the distance to the target based on the time-span between emitting the signal and receiving the echo.



Ultrasonic sensors are excellent at suppressing background interference. Virtually all materials which reflect sound can be detected, regardless of their colour. Even transparent materials or thin foils represent no problem for an ultrasonic sensor.

Microsonic ultrasonic sensors are suitable for target distances from 20 mm to 10 m and as they measure the time of flight, they can ascertain a measurement with pinpoint accuracy. Some of our sensors can even resolve the signal to an accuracy of 0.025 mm. Ultrasonic sensors can see through dust-laden air and ink mists. Even thin deposits on the sensor membrane do not impair its function.

Timing Diagram of Ultrasonic Sensor



Reference: <https://robu.in/ultrasonic-sensor-working-principle/>

- First, need to transmit trigger pulse of at least 10 us to the HC-SR04 Trig Pin.
- Then the HC-SR04 automatically sends Eight 40 kHz sound wave and wait for rising edge output at Echo pin.
- When the rising edge capture occurs at Echo pin, start the Timer and wait for a falling edge on Echo pin.
- As soon as the falling edge captures at the Echo pin, read the count of the Timer. This time count is the time required by the sensor to detect an object and return back from an object.

How to calculate Distance?

If you need to measure the specific distance from your sensor, this can be calculated based on this formula:

We know that, **Distance= Speed* Time**. The speed of sound waves is 343 m/s. So,

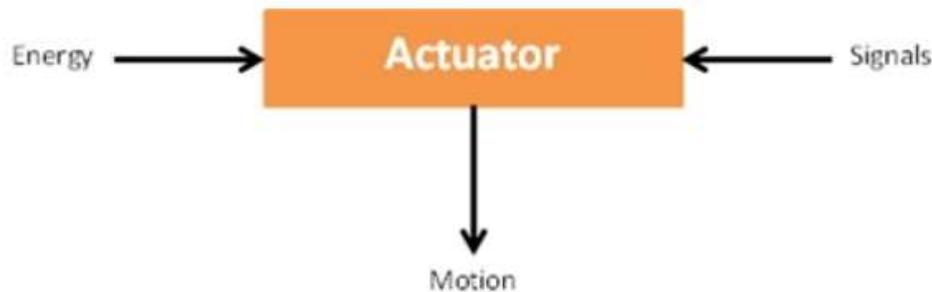
$$\text{Total Distance} = (343 * \text{Time of hight (Echo) pulse})/2$$

Total distance is divided by 2 because the signal travels from HC-SR04 to object and returns to the module HC-SR-04.

2.7 Introduction to Actuators

An actuator is a device that produces a motion by converting energy and signals going into the system. The motion it produces can be either rotary or linear.

An actuator is a device that produces a motion by converting energy and signals going into the system. The motion it produces can be either rotary or linear. Linear actuators, as the name implies, produce linear motion. This means that linear actuators can move forward or backwards on a set linear plane – a set distance they can travel in either direction before they must stop. Rotary actuators on the other hand produce rotary motion, meaning that the actuator revolves on a circular plane. Unlike the linear actuator, the rotary actuator is not limited by a set path, which means it can keep rotating in the same direction for as long as necessary.



Reference: <https://www.reac-group.com/media/1576/actuator-system.jpg?width=484&height=151>

Linear or rotary actuators are available in various forms depending on the power-supply source. The actuator could be electrical, pneumatic or hydraulic.

2.7.1 Types of Actuators

1. Hydraulic Actuators

A hydraulic actuator uses hydraulic power to perform a mechanical operation. They are actuated by a cylinder or fluid motor. The mechanical motion is converted to rotary, linear, or oscillatory motion, according to the need of the IoT device. Construction equipment uses hydraulic actuators because hydraulic actuators can generate a large amount of force.

Advantages:

- Hydraulic actuators can produce a large magnitude of force and high speed.
- Used in welding, clamping, etc.
- Used for lowering or raising the vehicles in car transport carriers.

2. Pneumatic Actuators

A pneumatic actuator uses energy formed by vacuum or compressed air at high pressure to convert into either linear or rotary motion. Example- Used in robotics, use sensors that work like human fingers by using compressed air.

Advantages:

- They are a low-cost option and are used at extreme temperatures where using air is a safer option than chemicals.
- They need low maintenance, are durable, and have a long operational life.
- It is very quick in starting and stopping the motion.

3. Electrical Actuators

An electric actuator uses electrical energy, is usually actuated by a motor that converts electrical energy into mechanical torque. An example of an electric actuator is a solenoid based electric bell.

Advantages:

- It has many applications in various industries as it can automate industrial valves.
- It produces less noise and is safe to use since there are no fluid leakages.
- It can be re-programmed and it provides the highest control precision positioning.

1. Relay

A Relay is a simple electromechanical switch. While we use normal switches to close or open a circuit manually, a Relay is also a switch that connects or disconnects two circuits. But instead of a manual operation, a relay uses an electrical signal to control an electromagnet, which in turn connects or disconnects another circuit.

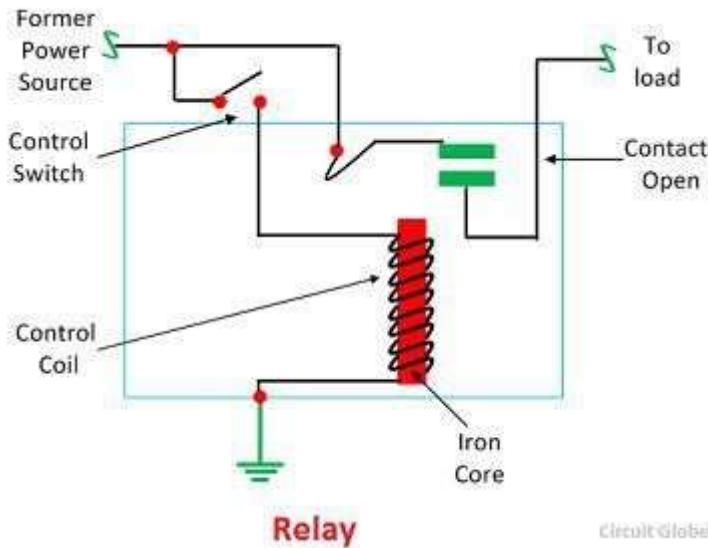


Reference: <https://files.seeedstudio.com/wiki/Grove-Relay/img/Thumbnail.jpg>

Working Principle of Relay

It works on the principle of an electromagnetic attraction. When the circuit of the relay senses the fault current, it energises the electromagnetic field which produces the temporary magnetic field.

This magnetic field moves the relay armature for opening or closing the connections. The small power relay has only one contacts, and the high power relay has two contacts for opening the switch.



Reference:

https://cdn.shopify.com/s/files/1/0559/1970/6265/files/Construction_of_relay_480x480.png?v=1667812202

The inner section of the relay is shown in the figure below. It has an iron core which is wound by a control coil. The power supply is given to the coil through the contacts of the load and the control switch. The current flows through the coil produces the magnetic field around it.

Due to this magnetic field, the upper arm of the magnet attracts the lower arm. Hence close the circuit, which makes the current flow through the load. If the contact is already closed, then it moves oppositely and hence open the contacts.

Relay Applications

Relays are used to protect the electrical system and to minimize the damage to the equipment connected in the system due to over currents/voltages. The relay is used for the purpose of protection of the equipment connected with it.

These are used to control the high voltage circuit with low voltage signal in applications audio amplifiers and some types of modems.

These are used to control a high current circuit by a low current signal in the applications like starter solenoid in automobile. These can detect and isolate the faults

that occurred in power transmission and distribution system. Typical application areas of the relays include

- Lighting control systems
- Telecommunication
- Industrial process controllers
- Traffic control
- Motor drives control
- Protection systems of electrical power system
- Computer interfaces
- Automotive
- Home appliances

2. LCD

In LCD 16x2, the term LCD stands for Liquid Crystal Display that uses a plane panel display technology, used in screens of computer monitors & TVs, smartphones, tablets, mobile devices, etc. Both the displays like LCD & CRTs look the same but their operation is different. Instead of electrons diffraction at a glass display, a liquid crystal display has a backlight that provides light to each pixel that is arranged in a rectangular network.

Every pixel includes a blue, red, green sub-pixel that can be switched ON/OFF. Once all these pixels are deactivated, then it will appear black and when all the sub-pixels are activated then it will appear white. By changing the levels of each light, different color combinations are achievable. This article discusses an overview of LCD 16X2 & its working with applications.

What is LCD 16X2?

An electronic device that is used to display data and the message is known as LCD 16x2. As the name suggests, it includes 16 Columns & 2 Rows so it can display 32 characters ($16 \times 2 = 32$) in total & every character will be made with 5×8 (40) Pixel Dots. So the total pixels within this LCD can be calculated as 32×40 otherwise 1280 pixels.



Reference: <https://www.watelectronics.com/lcd-16x2/>

16 X2 displays mostly depend on multi-segment LEDs. There are different types of displays available in the market with different combinations such as 8x2, 8x1, 16x1, and 10x2, however, the LCD 16x2 is broadly used in devices, DIY circuits, electronic projects due to less cost, programmable friendly & simple to access.

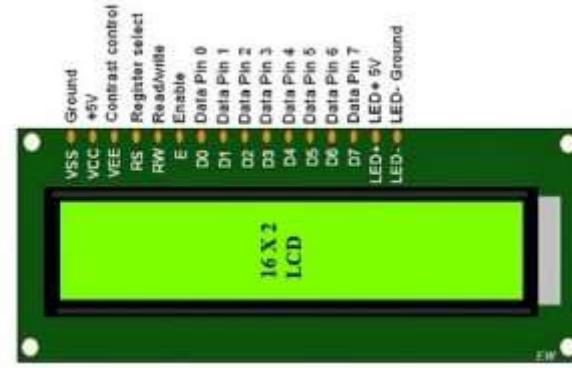
Specifications of LCD 16X2

The **specifications of LCD 16X2** are discussed below.

1. The operating voltage of this display ranges from 4.7V to 5.3V
2. The display bezel is 72 x 25mm
3. The operating current is 1mA without a backlight
4. PCB size of the module is 80L x 36W x 10H mm
5. HD47780 controller
6. LED color for backlight is green or blue
7. Number of columns – 16
8. Number of rows – 2
9. Number of LCD pins – 16
10. Characters – 32
11. It works in 4-bit and 8-bit modes
12. Pixel box of each character is 5x8 pixel
13. Font size of character is 0.125Width x 0.200height

LCD 16X2 Pin Configuration

The **pin configuration of LCD 16 X 2** is discussed below so that LCD 16x2 connection can be done easily with external devices.



16X2 LCD Pin Diagram

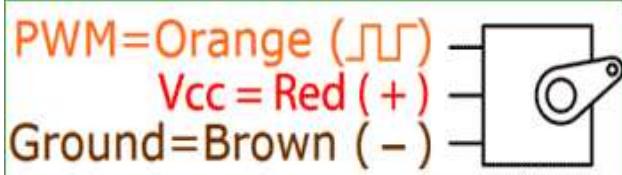
- Pin1 (Ground): This pin connects the ground terminal.
- Pin2 (+5 Volt): This pin provides a +5V supply to the LCD
- Pin3 (VE): This pin selects the contrast of the LCD.
- Pin4 (Register Select): This pin is used to connect a data pin of an MCU & gets either 1 or 0. Here, data mode = 0 and command mode =1.

- Pin5 (Read & Write): This pin is used to read/write data.
- Pin6 (Enable): This enables the pin must be high to perform the Read/Write procedure. This pin is connected to the data pin of the microcontroller to be held high constantly.
- Pin7 (Data Pin): The data pins are from 0-7 which are connected through the microcontroller for data transmission. The LCD module can also work on the 4-bit mode through working on pins 1, 2, 3 & other pins are free.
- Pin8 – Data Pin 1
- Pin9 – Data Pin 2
- Pin10 – Data Pin 3
- Pin11 – Data Pin 4
- Pin12 – Data Pin 5
- Pin13 – Data Pin 6
- Pin14 – Data Pin 7
- Pin15 (LED Positive): This is a +Ve terminal of the backlight LED of the display & it is connected to +5V to activate the LED backlight.
- Pin16 (LED Negative): This is a -Ve terminal of a backlight LED of the display & it is connected to the GND terminal to activate the LED backlight.

3. What is a Servo Motor?

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a **servo mechanism**. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. For this tutorial, we will be discussing only about the **DC servo motor working**. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc.

Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity. The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.



Reference [https://circuitdigest.com/article/servo-motor-working-and-basics#:~:text=Servo%20motor%20works%20on%20PWM,\(potentiometer\)%20and%20some%20gears.](https://circuitdigest.com/article/servo-motor-working-and-basics#:~:text=Servo%20motor%20works%20on%20PWM,(potentiometer)%20and%20some%20gears.)

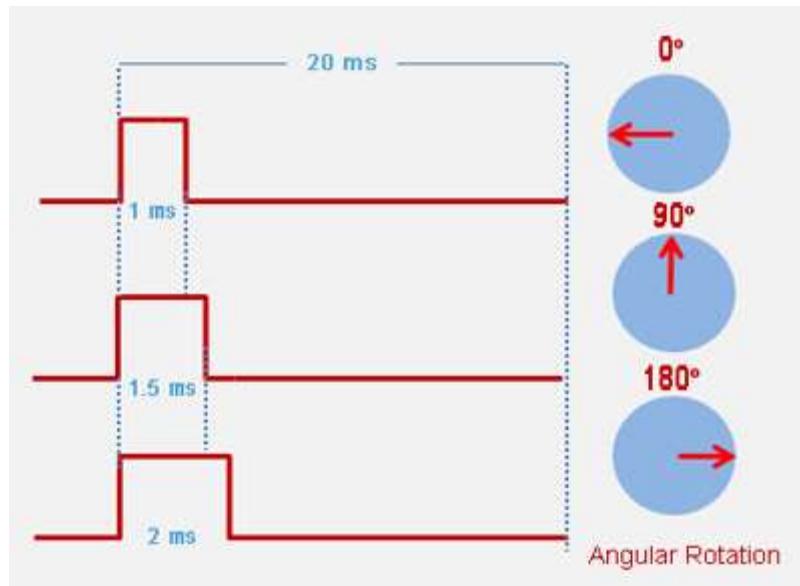
:

Controlling Servo Motor:

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU.

Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction from its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears**. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.



Reference: <https://circuitdigest.com/sites/default/files/inlineimages/servo-rotation.gif>

Servo motor can be rotated from 0 to 180 degrees, but it can go up to 210 degrees, depending on the manufacturing. This degree of rotation can be controlled by applying the **Electrical Pulse** of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. The pulse of 1 ms (1 millisecond) width can rotate the servo to 0 degrees, 1.5ms can rotate to 90 degrees (neutral position) and 2 ms pulse can rotate it to 180 degree.

All servo motors work directly with your +5V supply rails but we have to be careful about the amount of current the motor would consume if you are planning to use more than two servo motors a proper servo shield should be designed.

Unit 3: IoT Protocol and Cloud integration

Learning Outcomes:

- Understand different types of IoT Protocols
- Able to configure Device for IoT application
- Understand the Concept of Different Networking Devices

3.1 Introduction to Networking devices

3.1.1 What are network devices?

Network devices, or networking hardware, are physical devices that are required for communication and interaction between hardware on a computer network.

3.1.2 Types of network devices

Here is the common network device list:

- Hub
- Switch
- Router
- Bridge
- Gateway
- Modem
- Repeater
- Access Point

Repeater

A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2-port device.

Hub

A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, the collision domain of all hosts connected through Hub remains one. Also, they do not have the intelligence to find out the best path for data packets which leads to inefficiencies and wastage.

Types of Hubs

Active Hub: - These are the hubs that have their own power supply and can clean, boost, and relay the signal along with the network. It serves both as a repeater as well as a wiring center. These are used to extend the maximum distance between nodes.

Passive Hub: - These are the hubs that collect wiring from nodes and power supply from the active hub. These hubs relay signals onto the network without cleaning and boosting them and can't be used to extend the distance between nodes.

Intelligent Hub: - It works like active hubs and includes remote management capabilities. They also provide flexible data rates to network devices. It also enables an administrator to monitor the traffic passing through the hub and to configure each port in the hub.

Bridge

A bridge operates at the data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

Types of Bridges

Transparent Bridges: - These are the bridge in which the stations are completely unaware of the bridge's existence i.e. whether or not a bridge is added or deleted from the network, reconfiguration of the stations is unnecessary. These bridges make use of two processes i.e. bridge forwarding and bridge learning.

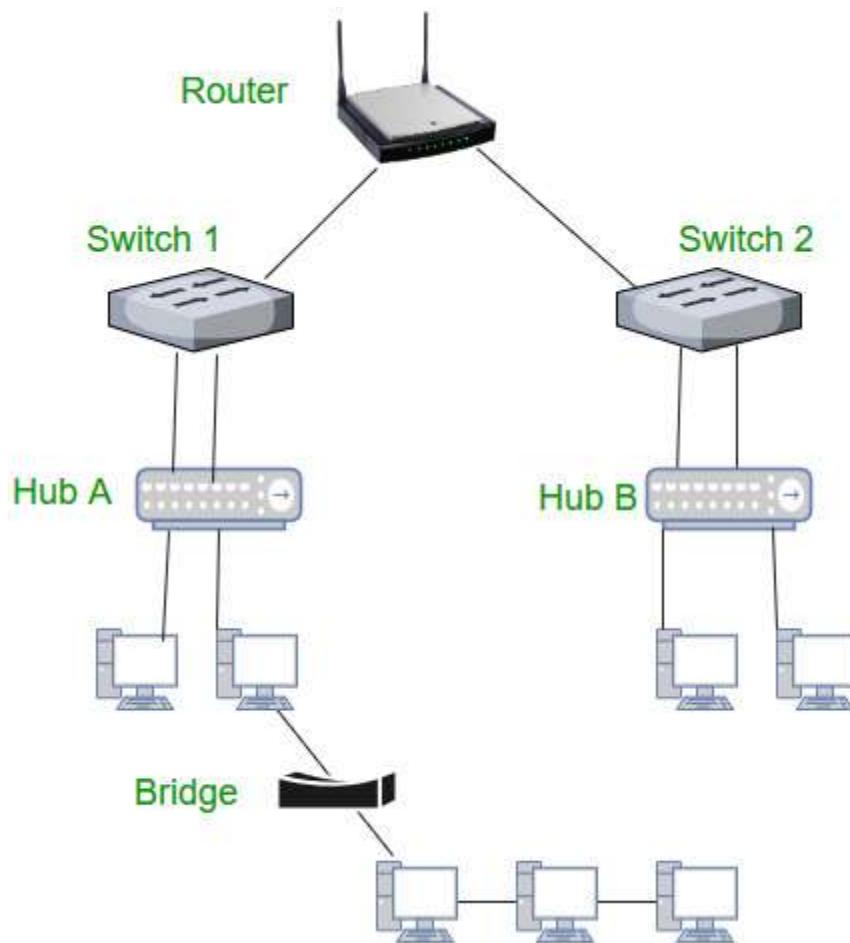
Source Routing Bridges: - In these bridges, routing operation is performed by the source station and the frame specifies which route to follow. The host can discover the frame by sending a special frame called the discovery frame, which spreads through the entire network using all possible paths to the destination.

Switch

A switch is a multiport bridge with a buffer and a design that can boost its efficiency (a large number of ports imply less traffic) and performance. A switch is a data link layer device. The switch can perform error checking before forwarding data, which makes it very efficient as it does not forward packets that have errors and forward good packets selectively to the correct port only. In other words, the switch divides the collision domain of hosts, but broadcast domain remains the same.

Routers

A router is a device like a switch that routes data packets based on their IP addresses. The router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.



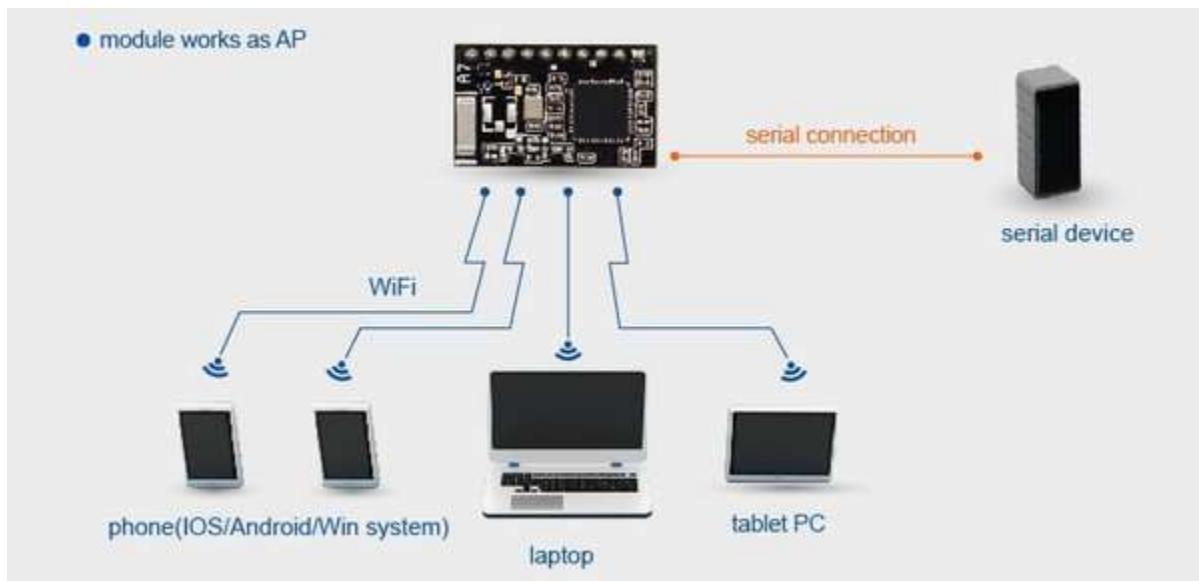
Reference: <https://www.geeksforgeeks.org/network-devices-hub-repeater-bridge-switch-router-gateways/>

Gateway

A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically work as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switches or routers. Gateway is also called a protocol converter.

3.2 Local and Personal Area Networks (LAN/PAN) for IOT

Networks that cover fairly short distances are called personal area networks (PAN) and local area networks (LAN). PAN and LAN networks are considered to be fairly cost-effective, but the transfer of data can sometimes be unreliable.



Reference: <https://www.quora.com/How-does-the-Internet-of-Things-work-in-a-LAN-network>

Wireless personal and local area network technologies that are commonly incorporated into IoT connectivity solutions are WiFi and Bluetooth. WiFi can be used for applications that run in a local environment, or in a distributed setting if there are multiple access points integrated into a larger network. One downside to WiFi is that it works only if the signal is strong and you're close to the access point. Also, WiFi is generally more power-hungry than people think, but it is possible to operate it in a way that's a little more power-efficient (for example, your device only connects periodically to send data, then goes back to sleep).

Bluetooth Low Energy (BLE) is a more energy-efficient wireless network protocol—if you're not receiving data constantly, a single battery running BLE could last up to five

years. However, compared to WiFi it is slower to transmit and is more limited in the amount of data it is capable of sending.

Both WiFi and Bluetooth are easy to connect in most cases, although WiFi does have some security challenges that may be difficult to overcome.

3.3 IOT WAN

A wide area network (also known as WAN), is a large network of information that is not tied to a single location. WANs can facilitate communication, the sharing of information and much more between devices from around the world through a WAN provider.

WANs can be vital for international businesses, but they are also essential for everyday use, as the internet is considered the largest WAN in the world.

An Internet of Things (IoT) gateway is a device which serves as the connection point between IoT devices and the cloud. This gateway can be a hardware appliance or virtual.

3.4 IOT NODE



Reference: [https://advdownload.advantech.com/productfile/PIS/WISE-1020/Product%20-%20Photo\(Main\)/WISE-1020-0S01E_3D_B20151008140951.jpg](https://advdownload.advantech.com/productfile/PIS/WISE-1020/Product%20-%20Photo(Main)/WISE-1020-0S01E_3D_B20151008140951.jpg)

The most numerous type of device in the IoT can be referred to as the node. These are all the exciting devices that are providing sensor data, or devices that are being

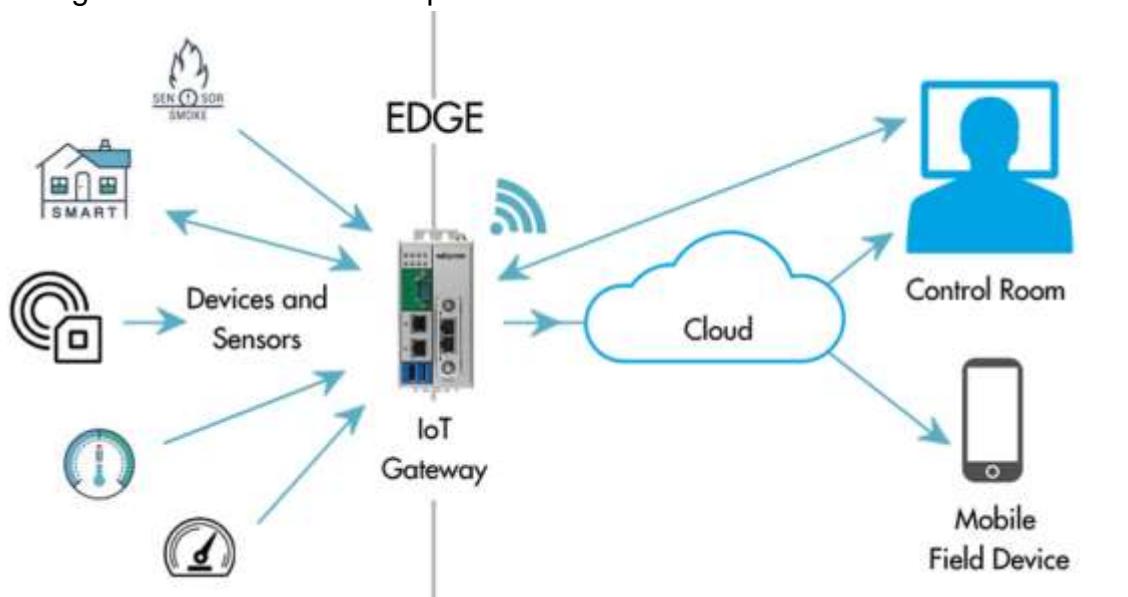
controlled from the cloud. This means things like door locks, security sensors, temperature sensors, and more.

Put simply, the node is the “thing” in Internet of Things, and until recently they were a practical impossibility. Nodes tend to be either lightweight sensor devices, which primarily gather status information over a pre-programmed interval, or middleweight devices which also offer controllable functions (like a door lock which can be toggled, traffic lights whose patterns can be adjusted, or industrial equipment which can be disabled if a fault is triggered).

The IoT node as we know it today, in its most minimal use case, can be a sensor embedded in an object that is never serviced again across the life of the device. They can be wireless and operated on a coin cell battery for years. What seemed impossible just a few years ago is now quickly becoming standard. And that’s thanks to incredible innovations in low-power operation of wireless modules.

3.5 IOT Gateway

An IoT gateway works by receiving data from IoT sensors, which it can then send onwards to the cloud; it also receives information from the cloud which then goes to the device itself to help it perform necessary functions, such as regulating environmental changes and detecting possible issues with functioning. All information moving from an IoT device to the cloud, or vice versa, goes through the connected IoT gateway. By managing this connection, the gateway can perform security tasks, help manage devices and translate protocols.



Reference: https://www.ezenroute.com/assets/images/gif/iot_gateway_img_1.jpg

One benefit of an IoT gateway is added security for the IoT network and data. Because the gateway protects information moving in both directions, it protects data moving to

the cloud from leaks, as well as prevents unauthorized control of IoT devices from outside parties.

Traditional IoT gateways are non-intelligent and perform basic gateway functionalities. However, non-intelligent gateways have recently been pushed out by "smart" IoT gateways, which are able to perform edge analytics on data produced by IoT devices before it is sent to the cloud. This makes analytics much faster and cuts down on storage for the vast amount of data produced by IoT products. However, performing edge analytics instead of keeping all IoT data may not be an effective solution in some situations, as this process causes the loss of a lot of raw data.

IoT gateways can also be used to convert non-cloud connected legacy devices to the internet for brownfield development. By connecting a gateway to a device's sensors, the data can be analyzed or transported directly by the gateway, even though the device itself would be unable to do so.

3.6 IPv4 vs IPv6

3.6.1 What is IP?

An IP (Internet Protocol) address is a numerical label assigned to each device connected to a computer network that uses the IP protocol for communication. An IP address acts as an identifier for a specific device on a particular network. The IP address is also called an IP number or Internet address.

IP address specifies the technical format of the addressing and packets scheme. Most networks combine IP with a TCP (Transmission Control Protocol). It also allows developing a virtual connection between a destination and a source.

Now in this IPv4 and IPv6 difference tutorial, we will learn What is IPv4 and IPv6?



Example: 127.255.255.255

Example:

2001:0db8:85a3:0000:0000:8a2e:0370:7334

Reference: <https://www.guru99.com/difference-ipv4-vs-ipv6.html>

What is IPv4?

IPv4 is an IP version widely used to identify devices on a network using an addressing system. It was the first version of IP deployed for production in the ARPANET in 1983. It uses a 32-bit address scheme to store 2^{32} addresses which is more than 4 billion

addresses. It is considered the primary Internet Protocol and carries 94% of Internet traffic.

What is IPv6?

IPv6 is the most recent version of the Internet Protocol. This new IP address version is being deployed to fulfill the need for more Internet addresses. It was aimed to resolve issues that are associated with IPv4. With 128-bit address space, it allows 340 undecillion unique address space. IPv6 is also called IPng (Internet Protocol next generation).

Internet Engineer Taskforce initiated it in early 1994. The design and development of that suite are now called IPv6.

KEY DIFFERENCE

- IPv4 is 32-Bit IP address whereas IPv6 is a 128-Bit IP address.
- IPv4 is a numeric addressing method whereas IPv6 is an alphanumeric addressing method.
- IPv4 binary bits are separated by a dot(.) whereas IPv6 binary bits are separated by a colon(:).
- IPv4 offers 12 header fields whereas IPv6 offers 8 header fields.
- IPv4 supports broadcast whereas IPv6 doesn't support broadcast.
- IPv4 has checksum fields while IPv6 doesn't have checksum fields
- When we compare IPv4 and IPv6, IPv4 supports VLSM (Variable Length Subnet Mask) whereas IPv6 doesn't support VLSM.
- IPv4 uses ARP (Address Resolution Protocol) to map to MAC address whereas IPv6 uses NDP (Neighbour Discovery Protocol) to map to MAC address.

IPv4	IPv6
IPv4 has a 32-bit address length	IPv6 has a 128-bit address length
It Supports Manual and DHCP address configuration	It supports Auto and renumbering address configuration
In IPv4 end to end, connection integrity is Unachievable	In IPv6 end to end, connection integrity is Achievable
It can generate 4.29×10^9 address space	Address space of IPv6 is quite large it can produce 3.4×10^{38} address space

IPv4	IPv6
The Security feature is dependent on application	IPSEC is an inbuilt security feature in the IPv6 protocol
Address representation of IPv4 is in decimal	Address Representation of IPv6 is in hexadecimal
Fragmentation performed by Sender and forwarding routers	In IPv6 fragmentation performed only by the sender
In IPv4 Packet flow identification is not available	In IPv6 packet flow identification are Available and uses the flow label field in the header
In IPv4 checksum field is available	In IPv6 checksum field is not available
It has broadcast Message Transmission Scheme	In IPv6 multicast and anycast message transmission scheme is available
In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided
IPv4 has a header of 20-60 bytes.	IPv6 has header of 40 bytes fixed
IPv4 consist of 4 fields which are separated by dot (.)	IPv6 consist of 8 fields, which are separated by colon (:)
Example of IPv4 – 66.94.29.13	Example of IPv6 – 2001:0000:3238:DFE1:0063:0000:0000:FEFB

3.7 Multi-homing

Multi-homing is a method of configuring one computer, called the host, with more than one network connection and IP address. The multi-homed method provides enhanced and reliable Internet connectivity without compromising efficient performance.

3.7.1 Why Is Multi-Homing Important?

With more and more Internet-connected devices, an organization's workforce is no longer sequestered to a single location. Instead, an organization may have employees connecting to their internal network and accessing sensitive data from across the globe. Because of this, old access security measures are no longer enough and must be replaced with safeguards that allow employees and other verified users safe and secure access from anywhere, at any time, from any device.

Using a multi-homed approach can:

- Help load balancing and let a network work with less downtime
- Give added safeguards against system failure
- Help maintain the system during disaster and recovery

Why multi-homing in IOT?

In IoT particular Node or an IoT device or the sub network, IoT sub network can be connected with multiple networks for improving the reliability. So, basically multi-homing is a concept that is used for improving the overall liability of the network in that way. So, in the same state if some component of the network or maybe a Node has gone down, there is another network that can take over.

3.8 IoT Protocol Stack

The Open Systems Interconnection (OSI) and Transmission Control Protocol/Internet Protocol (TCP/IP) networking models are the most common frameworks to encapsulate networking tasks in the form of multiple layers. Widely adopted IoT protocols can be mapped to these two models as outlined in the below table. For an IoT network to function effectively, protocols at different layers must be interoperable with each other.

OSI Model	TCP/IP Model	Layer functions	Protocols
Application	Application	Directly interact with and support user's applications	MQTT, HTTPS, AMQP, CoAP
Presentation			
Session			
Transport	Transport	Handle reliability, flow control, congestion avoidance, and error correction	TCP, UDP
Network	Internet	Involve logical addressing and define how data is routed from sources to final destination hosts identified by IP addresses (traffic directing)	IP (e.g. IPv6, 6LoWPAN...)
Data Link			
Physical	Network access & physical	Define the physical connection of end devices to the network	LPWAN, WiFi, LTE, BLE, Zigbee

Link Layer Protocols:

Link Layer protocols determine how the data is physically sent over the networks physical layer or medium(example copper wire, electrical cable, or radio wave). The Scope of The Link Layer is the Last Local Network connections to which host is attached. Host on the same link exchange data packets over the link layer using the link layer protocol. Link layer determines how the packets are coded and signaled by the hardware device over the medium to which the host is attached.

802.3 Ethernet:

802.3 is a collections of wired Ethernet standards for the link layer. For example 802.3 10BASE5 Ethernet that uses coaxial cable as a shared medium, 802.3.i is standard for 10 BASET Ethernet over copper twisted pair connection, Standards provide data rates from 10 Mb/s to 40 gigabits per second and the higher. The shared medium in Ethernet can be a coaxial cable , twisted pair wire or and Optical fiber. Shared medium carries the communication for all the devices on the network.

802.1- WI-FI:

IEEE 802.3 is a collections of wireless Local area network.(WLAN) communication standards, including extensive descriptions of the link layer. For example 802.11a operate in the 5 GHz band, 802.11b and 802.11g operate in the 2.4 GHz band. 802.11ac operate in the 5G hertz band.

802.16 wiMAX:

IEEE 802.16 is a collection of wireless broadband and Standards, including extensive descriptions for the link layer also called WiMAX wimax standard provides a data rates from from 1.5 Mb/s to 1Gb/s the recent update provides data rates of hundred megabits per second for mobile station.

802.15.4 LR-WPAN:

IEEE 802.15.4 is a collection of standards for low rate wireless personal area network (LRWPAN). These standards form the basis of specifications for high level communication like Zigbee. LR-WPAN standards provide data rates from 40 k b/s. These standards provide low cost and low speed communications for power constrained devices.

2G / 3G / 4G mobile communications:

These are the different generations of mobile communication standards including second generation (2G including GSM and CDMA), 3rd Generation (3G including UMTS and CDMA2000) and 4th generation 4G including LTE.

Network / internet layer Protocols:

The network layer is responsible for sending of IP datagrams from the source network to the destination network. This layer performs the host addressing and packet routing. The datagrams contain a source and destination address which are used to route them from the source to the destination across multiple networks. Host identification is done using the hierarchy IP addressing schemes such as IPv4 or IPv6.

IPv4:

Internet protocol version for open parents close (IPv4) is the most deployed internet protocol that is used to identify the device is on a network using a hierarchy address schemes. It uses 32-bit address scheme that allows a total of 2³² addresses. As more and more devices got connected to the internet, the IPv4 has succeeded by IPv6.

IPv6:

It is the newest version of internet protocol and successor to IPv4. IPv6 uses 128-bit address schemes that allow a total of 2¹²⁸ addresses.

6LoWPAN:

IPv6 over low power wireless personal area networks brings IP protocol to the low power device which has limited processing capability. It operates in the 2.4 GHz frequency range and provides the data transfer rate up to 50 kb/s.

Transport layer protocols:

The Transport layer protocols provide end-to-end message transfer capability independent of the underlying network. The message transfer capability can be set up on connections, either using handshake or without handshake acknowledgements.

Provides functions such as error control, segmentation, flow control and congestion control.

TCP:

Transmission control protocol is the most widely used to transport layer protocol that is used by the web browsers along with HTTP , HTTPS application layer protocols email program (SMTP application layer protocol) and file transfer protocol. TCP is a connection Oriented and stateful protocol while IP protocol deals with sending packets,TCP ensures reliable transmissions of packets in order. TCP also provide error deduction capability so that duplicate packets can be discarded and low packets are retransmitted The flow control capability ensures that the rate at which the sender since the data is now to too to high for the receiver to process.

UDP:

Unlike TCP, which requires carrying out an initial setup procedure, UDP is a connection less protocol. UDP is useful for time sensitive application they have very small data units to exchange and do not want the overhead of connection setup. UDP is a transactions oriented and stateless protocol. UDP does not provide guaranteed delivery, ordering of messages and duplicate eliminations.

Application layer protocols:

Application layer protocol define how the application interfaces with the lower layer protocols to send the data over the network. Data are typically in files, is encoded by the application layer protocol and encapsulated in the transport layer protocol. Application layer protocol enable process-to-process connection using ports.

Http:

Hypertext transfer protocol is the application layer protocol that forms the foundations of world wide web http includes, ,commands such as GET, PUT,POST, DELETE, HEAD, TRACE, OPTIONS etc. The protocol follows a request response model where are client sends request to server using the http, commands.Http is a stateless protocol and each http request is independent father request and http client can be a browser or an application running on the client example and application running on an IoT device ,mobile mobile applications or other software.

CoAP:

Constrained application protocol is an application layer protocol for machine to machine application M2M meant for constrained environment with constrained devices and constrained networks. Like http CoAP is a web transfer protocol and uses a request- response model, however it runs on the top of the UDP instead of TC CoAP uses a client –server architecture where client communicate with server using

connectionless datagrams. It is designed to easily interface with http like http, CoAP supports method such as GET, PUT, DELETE .

Websocket:

Websocket protocol allows full duplex communication over a single socket connection for sending message between client and server. Websocket is based on TCP and Allows streams of messages to be sent back and forth between the client and server while keeping the TCP connection open. The client can be a browser, a mobile application and IoT device

MQTT :

Message Queue Telemetry Transport it is a lightweight message protocol based on public -subscribe model MQTT uses a client server Architecture by the clients such as an IoT device connect to the server also called the MQTT broker and publishers' message to topic on the server. The broker forwards the message to the clients subscribed to topic MQTT is well suited for constrained environments.

AMQP:

Advanced Message Queuing protocols. it is an open application layer protocol for business messaging. AMQP support point to point and publish - subscribe model routing and queuing. AMQP broker receive message from publishers example devices or applications that generate data and about them over connections to consumers publishers publish the message to exchange which then distribute message copies to queues.

3.9 Types of Wireless Communication Protocols in IOT

Wi-Fi

Wi-Fi (Wireless Fidelity) is the most popular IOT communication protocols for wireless local area network (WLAN) that utilizes the IEEE 802.11 standard through 2.4 GHz UHF and 5 GHz ISM frequencies. Wi-Fi provides Internet access to devices that are within the range of about 20 - 40 meters from the source. It has a data rate upto 600 Mbps maximum, depending on channel frequency used and the number of antennas. In embedded systems, ESP series controllers from Espressif are popular for building IoT based Applications. ESP32 and ESP8266 are the most commonly use wifi modules for embedded applications.

In terms of using the Wi-Fi protocol for IOT, there are some pros & cons to be considered. The infrastructure or device cost for Wi-Fi is low & deployment is easy but the power consumption is high and the Wi-Fi range is quite moderate. So, the Wi-Fi may not be the best choice for all types of IOT applications but it can be used for applications like Home Automation.

There are many development boards available that allow people to build IOT applications using Wi-Fi. The most popular ones are the Raspberry Pi and Node MCU. These boards allow people to build IOT prototypes and also can be used for small real-time applications. Likewise is the Marvell Avastar 88W8997 SoC, which follows the Wi-Fi's IEEE 802.11n standard. The chip has applications like wearables, wireless audio & smart home.

Bluetooth

Bluetooth is a technology used for exchanging data wirelessly over short distances and preferred over various **IOT network protocols**. It uses short-wavelength UHF radio waves of frequency ranging from 2.4 to 2.485 GHz in the ISM band. The Bluetooth technology has 3 different versions based on its applications:

- **Bluetooth:** The Bluetooth that is used in devices for communication has many applications in IOT/M2M devices nowadays. It is a technology using which two devices can communicate and share data wirelessly. It operates at 2.4GHz ISM band and the data is split in packets before sending and then is shared using any one of the designated 79 channels operating at 1 MHz of bandwidth.
- **BLE (Bluetooth 4.0, Bluetooth Low Energy):** The BLE has a single main difference from Bluetooth that it consumes low power. With that, it makes the product of low cost & more long-lasting than Bluetooth.
- **iBeacon:** It is a simplified communication technique used by Apple and is completely based on Bluetooth technology. The Bluetooth 4.0 transmits an ID called UUID for each user and makes it easy to communicate between iPhone users.

Bluetooth has many applications, such as in telephones, tablets, media players, robotics systems, etc. The range of Bluetooth technology is between 50 – 150 meters and the data is being shared at a maximum data rate of 1 Mbps.

After launching the BLE protocol, there have been many new applications developed using Bluetooth in the field of IOT. They fall under the category of low-cost consumer products and Smart-Building applications. Like Wi-Fi, **Bluetooth also has a module Bluetooth HC-05 that can be interfaced with development boards like Arduino or Raspberry Pi to build DIY projects**. When it comes to Real-time applications, Marvell's Avastar 88W8977 comes with Bluetooth v4.2 and has features like high speed, mesh networking for IOT. Another product, M5600 is a wireless pressure transducer with a Bluetooth v4.0 embedded in it.

3.10 Cloud Integration IoT services

1. ThingSpeak for IoT

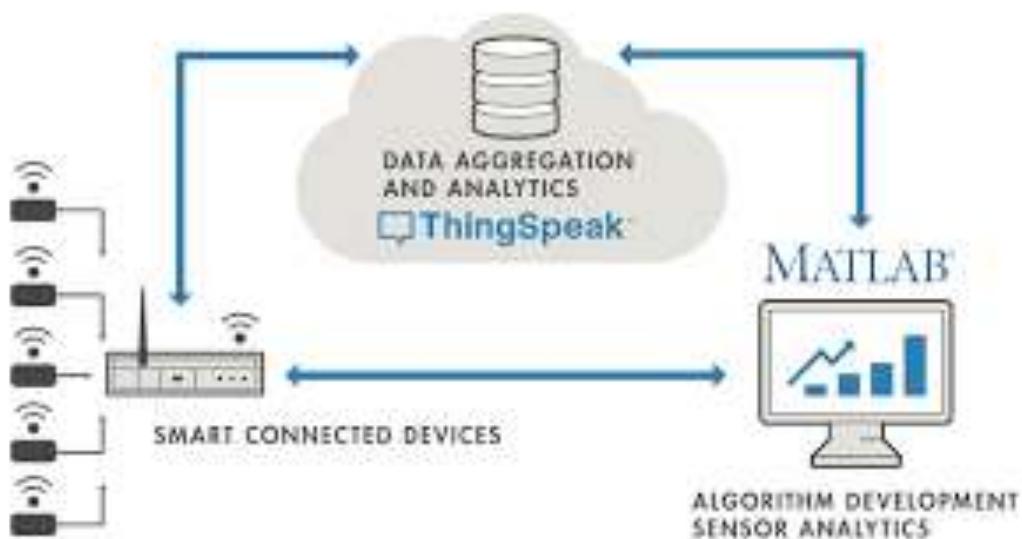
ThingSpeak™ is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics.

Learn More About ThingSpeak

Internet of Things (IoT) describes an emerging trend where a large number of embedded devices (things) are connected to the Internet. These connected devices communicate with people and other things and often provide sensor data to cloud storage and cloud computing resources where the data is processed and analyzed to gain important insights. Cheap cloud computing power and increased device connectivity is enabling this trend.

IoT solutions are built for many vertical applications such as environmental monitoring and control, health monitoring, vehicle fleet monitoring, industrial monitoring and control, and home automation.

At a high level, many IoT systems can be described using the diagram below:



Reference:

https://www.mathworks.com/products/thingspeak/_jcr_content/mainParsys/band_copy_copy_copy/mainParsys/columns/2/image.adapt.full.medium.svg/1636629884511.svg

On the left, we have the smart devices (the “things” in IoT) that live at the edge of the network. These devices collect data and include things like wearable devices, wireless temperatures sensors, heart rate monitors, and hydraulic pressure sensors, and machines on the factory floor.

In the middle, we have the cloud where data from many sources is aggregated and analyzed in real time, often by an IoT analytics platform designed for this purpose.

The right side of the diagram depicts the algorithm development associated with the IoT application. Here an engineer or data scientist tries to gain insight into the collected data by performing historical analysis on the data. In this case, the data is pulled from the IoT platform into a desktop software environment to enable the engineer or scientist to prototype algorithms that may eventually execute in the cloud or on the smart device itself.

An IoT system includes all these elements. ThingSpeak fits in the cloud part of the diagram and provides a platform to quickly collect and analyze data from internet connected sensors.

ThingSpeak Key Features

ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. Some of the key capabilities of ThingSpeak include the ability to:

- Easily configure devices to send data to ThingSpeak using popular IoT protocols.
- Visualize your sensor data in real-time.
- Aggregate data on-demand from third-party sources.
- Use the power of MATLAB to make sense of your IoT data.
- Run your IoT analytics automatically based on schedules or events.
- Prototype and build IoT systems without setting up servers or developing web software.
- Automatically act on your data and communicate using third-party services like Twilio® or Twitter®.

Lets start work with ThingSpeak.

Step 1: Click on this link <https://thingspeak.com/>



Step 2: Create Account on the ThingSpeak



Step 3: Create new Channel

The screenshot shows the 'My Channels' section of the ThingSpeak website. A single channel is listed:

- Name:** Ultrasonic data
- Created:** 2023-11-01 05:25
- Updated:** 2023-11-01 05:25

Below the channel list are links for **Home**, **Alerts**, **Settings**, **Sharing**, **API Keys**, **Data Import / Export**.

Help section contains links to:

- Collect data in a ThingSpeak channel from a device, store another channel, or from the web.
- Click New Channel to create a new ThingSpeak channel.
- Click on the column headers of the table to sort by the metrics in that column or click on a tag to filter channels with that tag.
- Learn more about [ThingSpeak Channels](#) and [How to Buy](#).
- Learn more about [ThingSpeak Channel](#).

Examples section lists:

- [Rain](#)
- [Arduino MRR100](#)
- [ESP8266](#)
- [Raspberry Pi](#)
- [Infrared PIR](#)

Upgrade section:

- Need to send more data? [Get Started!](#)
- Need to use more data? [Get Started!](#)

Step 4: Select the appropriate settings

The screenshot shows the 'Channel Settings' page for the 'Ultrasonic data' channel. The 'Channel Settings' tab is selected.

Channel Settings:

- Percentage complete: 30%
- Channel ID: 2012607
- Name:** Ultrasonic data
- Description:** (empty)
- Field 1:** Distance
- Field 2:** (empty)
- Field 3:** (empty)
- Field 4:** (empty)

Help section contains links to:

- Percentage complete: Calculated based on data received from the various fields of a channel. Since the device, location, location code, notes, and tags do not contribute to the percentage complete.
- Channel Name: Enter a unique name for the ThingSpeak channel.
- Description: Enter a description of the ThingSpeak channel.
- Field: Click the link to enable the field, and enter a field name. Each ThingSpeak channel can have up to 4 fields.
- Metadata: Enter information about channel fields, including `2020/01/01` as `EDD` date.
- Tags: Enter keywords that identify the channel. Separate tags with commas.
- Click External Site: If you have a website that contains information about your ThingSpeak channel, specify the URL.

Step 5: Generate API key and Copy key in your code

The screenshot shows the main channel page for the 'Ultrasonic data' channel. The 'API Keys' tab is selected.

Write API Key section:

- Key: 2FTDSF4eG5IC80Mj
- Generate New API Key** button

Read API Keys section:

- Key: E311988X9ME84LXA
- Get** button

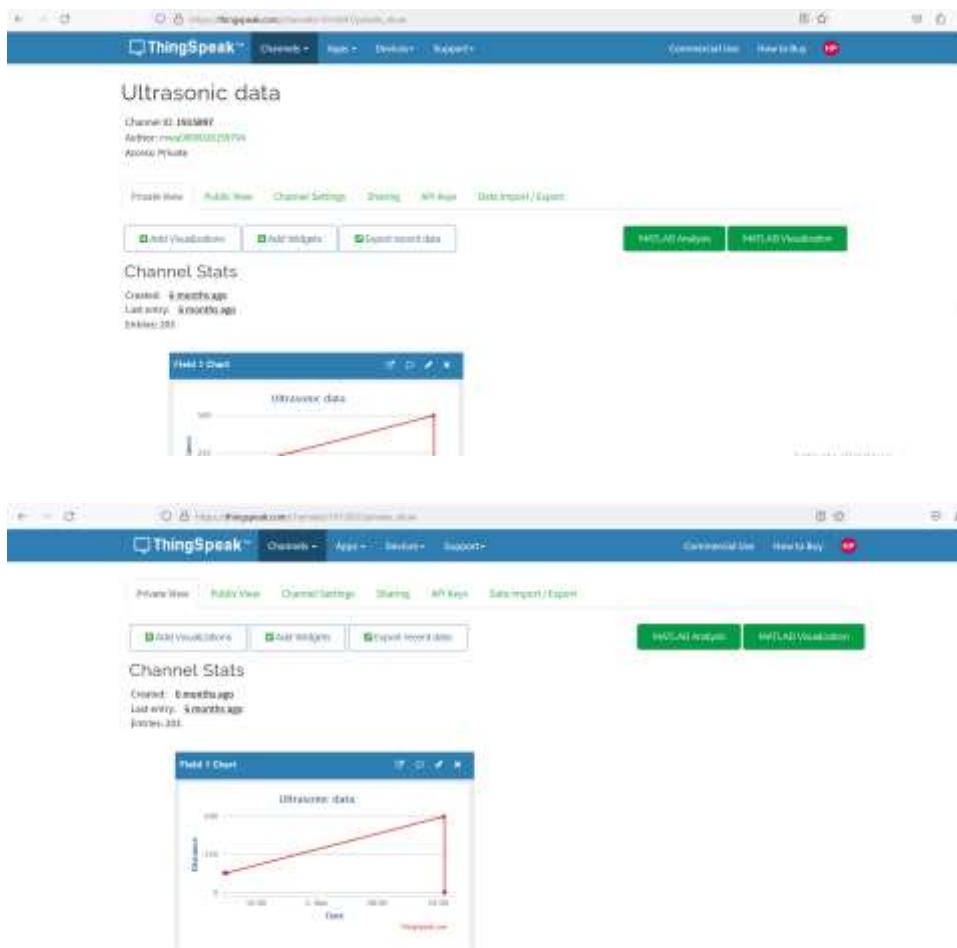
Help section:

- Write API Key: Use this key to write data to a channel. If you feel your key has been compromised, click [Delete](#) to generate a new API Key.
- Read API Key: This key is used to allow others to view your private channel's data and metrics. Click [Generate New Read API Key](#) to generate an additional read key for the channel.
- Notes: Use this field to enter information about channel and keys. For example, add notes to keep track of users with access to your channel.

API Requests section:

- Write a Channel Feed:** `POST https://api.thingspeak.com/update.php` `Content-Type: application/x-www-form-urlencoded`

Step 6: Run the code and see the results



2. Blynk framework

Blynk is an IoT platform for iOS or Android smartphones that is used to control Arduino, Raspberry Pi and NodeMCU via the Internet. This application is used to create a graphical interface or human machine interface (HMI) by compiling and providing the appropriate address on the available widgets.

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things.

There are three major components in the platform:

Blynk App: – It allows you to create amazing interfaces for your projects using various widgets which are provided.

Blynk Server: – It is responsible for all the communications between the smartphone and hardware.

You can use the Blynk Cloud or run your private Blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

Blynk Libraries: – It enables communication, for all the popular hardware platforms, with the server and process all the incoming and outgoing commands.

The process that occurs when someone presses the Button in the Blynk application is that the data will move to Blynk Cloud, where data magically finds its way to the hardware that has been installed. It works in the opposite direction and everything happens in a blink of an eye.



Reference:[119525085-e464a300-bd86-11eb-84dc-a4f3de0f7ec9.png \(3200x1632\) \(user-images.githubusercontent.com\)](https://user-images.githubusercontent.com/119525085/e464a300-bd86-11eb-84dc-a4f3de0f7ec9.png)

Unit 4: IoT Hardware and Software & implementation

Learning Outcomes:

- Understand the Hardware structure of Raspberry Pi
- Able to install operating system into raspberry pi
- Able to interface various sensors and actuators

4.1 Introduction to Raspberry Pi

Raspberry Pi, developed by Raspberry Pi Foundation in association with Broadcom, is a series of small single-board computers and perhaps the most inspiring computer available today.

Raspberry Pi is a small single board computer. By connecting peripherals like Keyboard, mouse, display to the Raspberry Pi, it will act as a mini personal computer. Raspberry Pi is popularly used for real time Image/Video Processing, IoT based applications and Robotics applications. Raspberry Pi is slower than laptop or desktop but is still a computer which can provide all the expected features or abilities, at a low power consumption.

Raspberry Pi Foundation officially provides Debian based Raspbian OS. Also, they provide NOOBS OS for Raspberry Pi. We can install several Third-Party versions of OS like Ubuntu, Archlinux, RISC OS, Windows 10 IOT Core, etc.

Raspbian OS is official Operating System available for free to use. This OS is efficiently optimized to use with Raspberry Pi. Raspbian have GUI which includes tools for Browsing, Python programming, office, games, etc.

We should use SD card (minimum 8 GB recommended) to store the OS (operating System). Raspberry Pi is more than computer as it provides access to the on-chip hardware i.e. GPIOs for developing an application. By accessing GPIO, we can connect devices like LED, motors, sensors, etc and can control them too.

It has ARM based Broadcom Processor SoC along with on-chip GPU (Graphics Processing Unit).

The CPU speed of Raspberry Pi varies from 700 MHz to 1.2 GHz. Also, it has on-board SDRAM that ranges from 256 MB to 1 GB.

Raspberry Pi also provides on-chip SPI, I2C, I2S and UART modules.

There are different versions of raspberry pi available as listed below:

- Raspberry Pi 1 Model A
- Raspberry Pi 1 Model A+
- Raspberry Pi 1 Model B
- Raspberry Pi 1 Model B+
- Raspberry Pi 2 Model B\
- Raspberry Pi 3 Model B
- Raspberry Pi Zero

Raspberry Pi 4 – model B

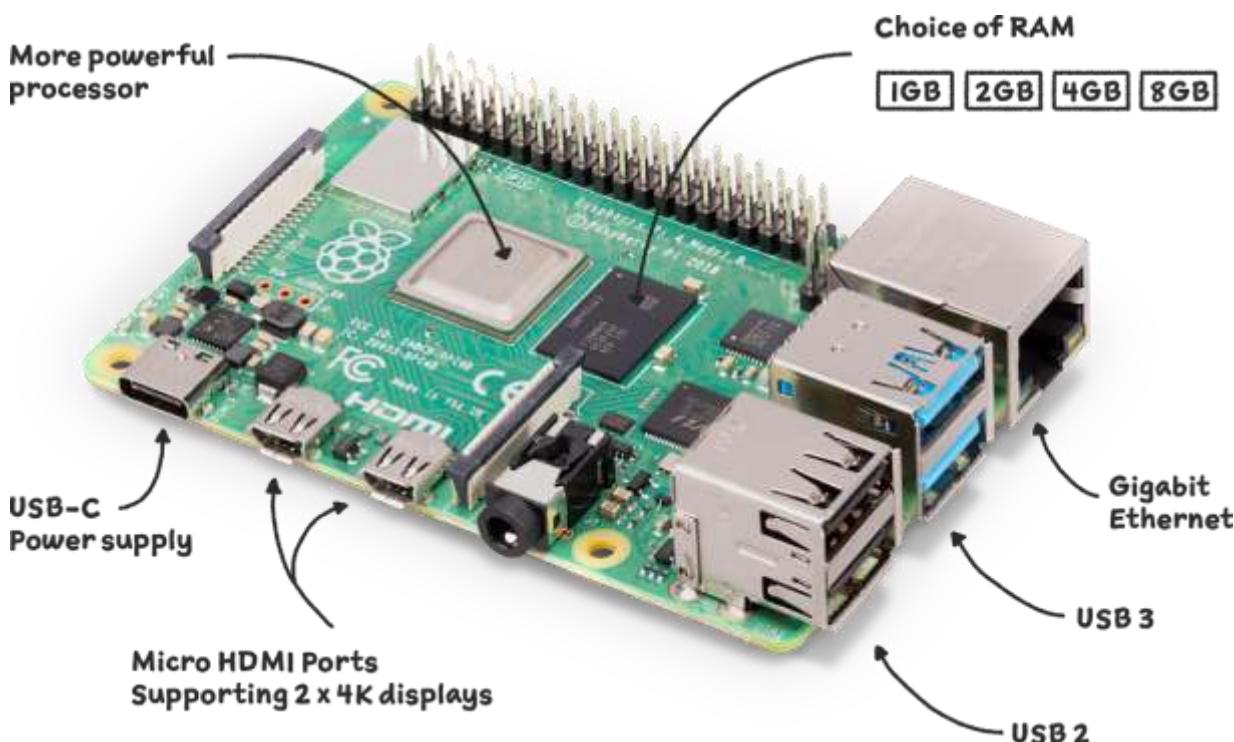


Fig: Raspberry Pi 4 – Model B
 Reference: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

Raspberry Pi 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+, while retaining backwards compatibility and similar power consumption. For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems.

This product's key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware

video decode at up to 4Kp60, up to 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on).

The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.

Specification

Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	1GB, 2GB, 4GB or 8GB LPDDR4 (depending on model) with on-die ECC
Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.
GPIO	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Video & Sound	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Multimedia	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics
SD Card Support	Micro SD card slot for loading operating system and data storage
Input Power	5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)-enabled (requires separate PoE HAT)
Environment	Operating temperature 0–50°C

Uses

Like a desktop computer, you can do almost anything with the Raspberry Pi. You can start and manage programs with its graphical windows desktop. It also has the shell for accepting text commands.

We can use the Raspberry Pi computer for the following –

- Playing games
- Browsing the internet
- Word processing
- Spreadsheets
- Editing photos

- Paying bills online
- Managing your accounts.

The best use of Raspberry Pi is to learn how a computer works. You can also learn how to make electronic projects or programs with it.

It comes with two programming languages, **Scratch** and **Python**. Through GPIO (general-purpose input output) pins, Raspberry Pi can be connected to other circuits, so that you can control the other devices of your choice.

4.2 Install Raspbian OS in Raspberry Pi 4 B

Raspberry Pi recommends the use of Raspberry Pi Imager to install an operating system on to your SD card. You will need another computer with an SD card reader to install the image. Raspberry Pi Imager can be run on another Raspberry Pi, but also works on Microsoft Windows, Apple macOS, and Linux.

Using Raspberry Pi Imager

Raspberry Pi have developed a graphical SD card writing tool that works on Mac OS, Ubuntu 18.04, and Windows called Raspberry Pi Imager; this is the easiest option for most users since it will download the image automatically and install it to the SD card.

Download the latest version of [Raspberry Pi Imager](#) and install it. If you want to use Raspberry Pi Imager from a second Raspberry Pi, you can install it from a terminal using ***sudo apt install rpi-imager***. Then:

- Connect an SD card reader with the SD card inside.
- Open Raspberry Pi Imager and choose the required OS from the list presented.
- Choose the SD card you wish to write your image to.
- Review your selections and click on the Write button to begin writing data to the SD Card.

Note:

If using Raspberry Pi Imager on Windows 10 with controlled folder access enabled, you will need to explicitly allow Raspberry Pi Imager permission to write the SD card. If this is not done, the imaging process will fail with a "failed to write"

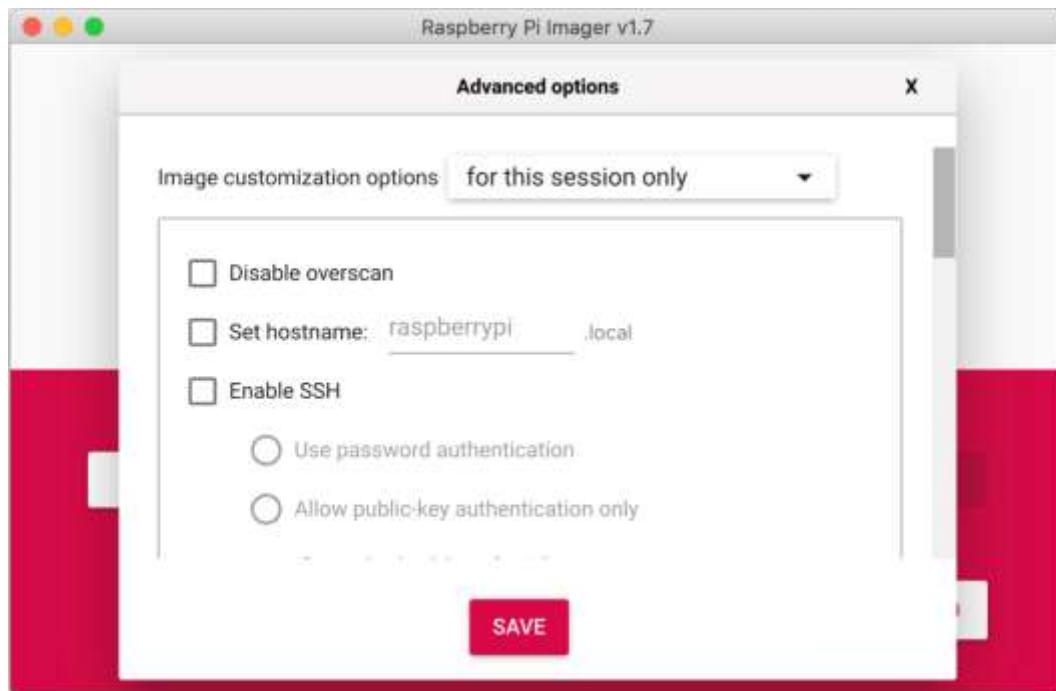
You can now insert the SD card into the Raspberry Pi and power it up. When your Raspberry Pi boots for the first time a configuration wizard will run that allows you to set up your Raspberry Pi.

Advanced Options

When you have the Raspberry Pi Imager open, and after you have selected the operating system to install, a cog wheel will appear allowing you to open an "Advanced Options" menu if it is supported by the operating system. This menu lets you carry out tasks like enabling SSH, or setting your Raspberry Pi's hostname, and configuring the default user before first boot.



Amongst other things the Advanced Options menu is useful for when you want to configure a headless Raspberry Pi.



If you are installing Raspberry Pi OS Lite and intend to run it headless, you will still need to create a new user account. Since you will not be able to create the user account on first boot, you **MUST** configure the operating system using the Advanced Menu.

Downloading an Image

If you are using a different tool than Raspberry Pi Imager to write to your SD Card, most require you to download the image first, then use the tool to write it to the card. Official images for recommended operating systems are available to download from the Raspberry Pi website downloads page. Alternative operating systems for Raspberry Pi computers are also available from some third-party vendors.

You may need to unzip the downloaded file (.zip) to get the image file (.img) you need to write to the card.

4.3 Configure GrovePi+ Kit

What is GrovePi+

GrovePi+ is add-on board with 15 Grove 4-pin interfaces that brings Grove sensors to the Raspberry Pi. It is the newest version compatible with Raspberry Pi model B/B+/A+/2/3/4 perfectly.

GrovePi+ is an easy-to-use and modular system for hardware hacking with the Raspberry Pi, no need for soldering or breadboards: plug in your Grove sensors and start programming directly. Grove is an easy to use collection of more than 100 inexpensive plug-and-play modules that sense and control the physical world. By connecting Grove Sensors to Raspberry Pi, it empowers your Pi in the physical world. With hundreds of sensors to choose from Grove families, the possibilities for interaction are endless.

- Compatible with Raspberry Pi model B/B+/A+/2/3/4
- Faster SPI and higher reliability UART connections
- Easier to assemble camera cables and LCD cables
- Simplified procedures of firmware update

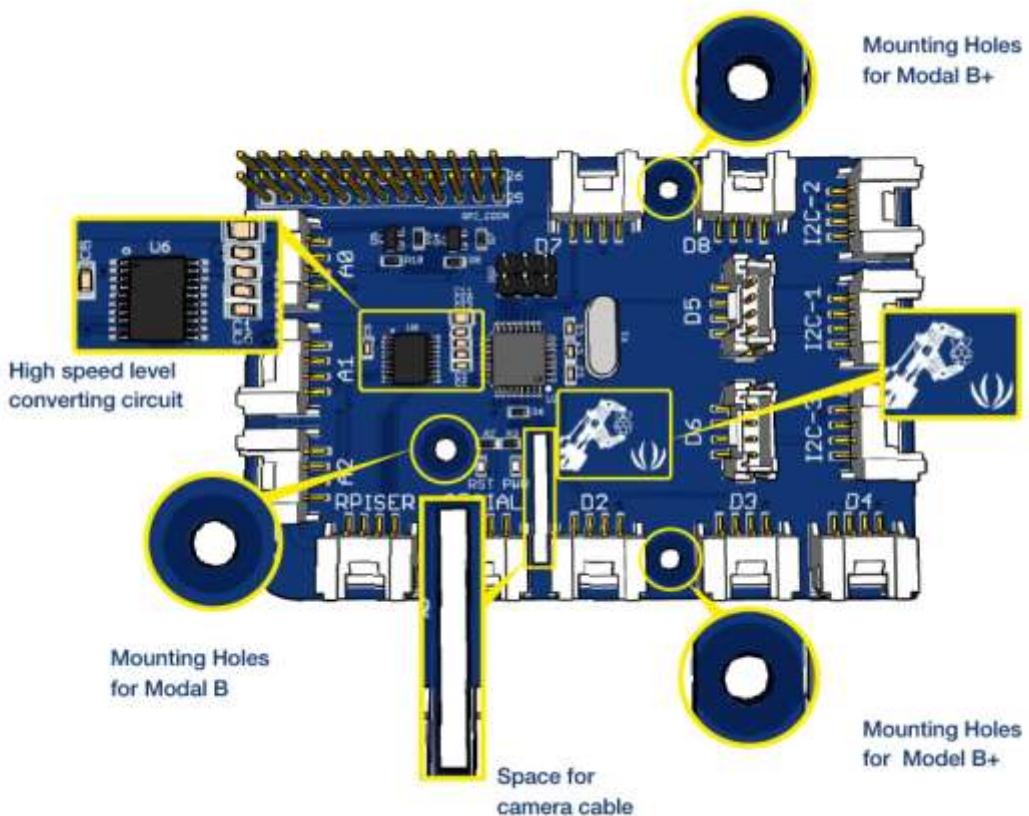


Fig: GrovePi+ Board

Reference: [GrovePi+ add-on for Raspberry Pi – Seeed Studio](#)

Set-up in 4 simple steps

- Step 1:** Slip the GrovePi+ board over your Raspberry Pi
- Step 2:** Connect the Grove modules to the GrovePi+ board
- Step 3:** Upload your program to Raspberry Pi
- Step 4:** Begin taking in the world data

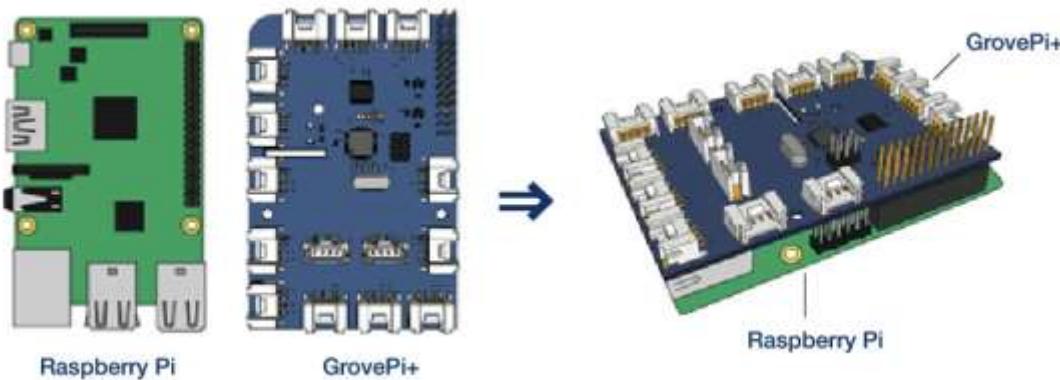


Fig: GrovePi+ Board and Raspberry Pi interface
 Reference: [GrovePi+ add-on for Raspberry Pi – Seeed Studio](#)

Features:

- 7 digital Ports
- 3 analoge Ports
- 3 I2C ports
- 1 Serial port connect to GrovePi
- 1 Serial port connect to Raspberry Pi
- Grove header Vcc output Voltage: 5Vdc

The GrovePi Starter Kit gets you up and running with the GrovePi quickly. The starter kit bundles the most popular sensors for education and hobbyists, and lets you start playing and prototyping hardware with Raspberry Pi. No soldering required!

The GrovePi Starter Kit package includes:

- GrovePi+ Board
- 12 different Grove sensors and modules
- Grove cables for connecting the sensors to the GrovePi board.

GrovePi Kit Includes



Grove pi+ × 1



Grove - sound Sensor × 1



Grove - Temperature&Humidity × 1



Grove - light sensor × 1



Grove - Relay × 1



Grove - Button × 1



Grove - Ultrasonic Ranger × 1



Grove - Rotary Angle Sensor × 1



Grove-LCD RGB Backlight × 1



Grove - red led × 1



Grove Buzzer × 1



Grove blue led × 1



Grovepi+Guidebook × 1

Cables × 10



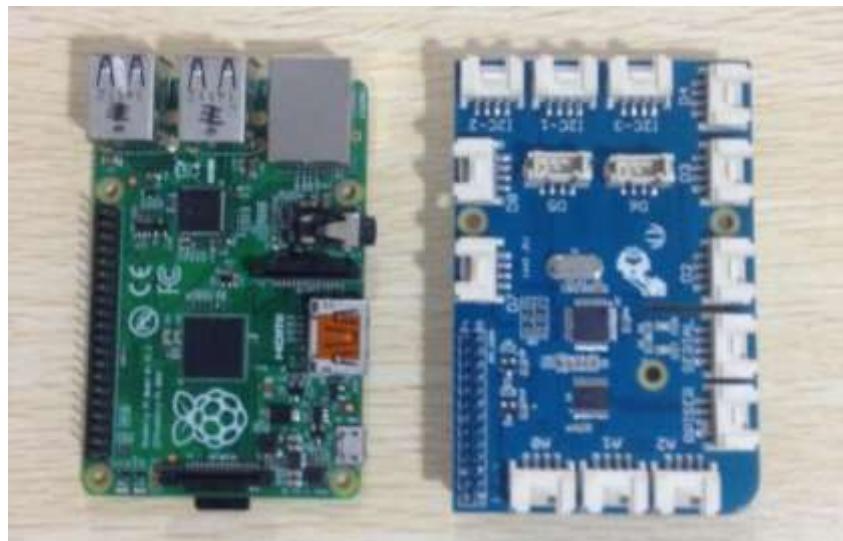
Grove green led × 1

Fig: Grove Pi Kit Sensors

Reference: [starter-Kit-content.jpg \(758x647\) \(dexterindustries.com\)](https://dexterindustries.com/starter-Kit-content.jpg)

Hardware connection for GrovePi+ to raspberry Pi

First, mount your GrovePi on the Raspberry Pi. The GrovePi slides over top of the Raspberry Pi as shown in the picture below.



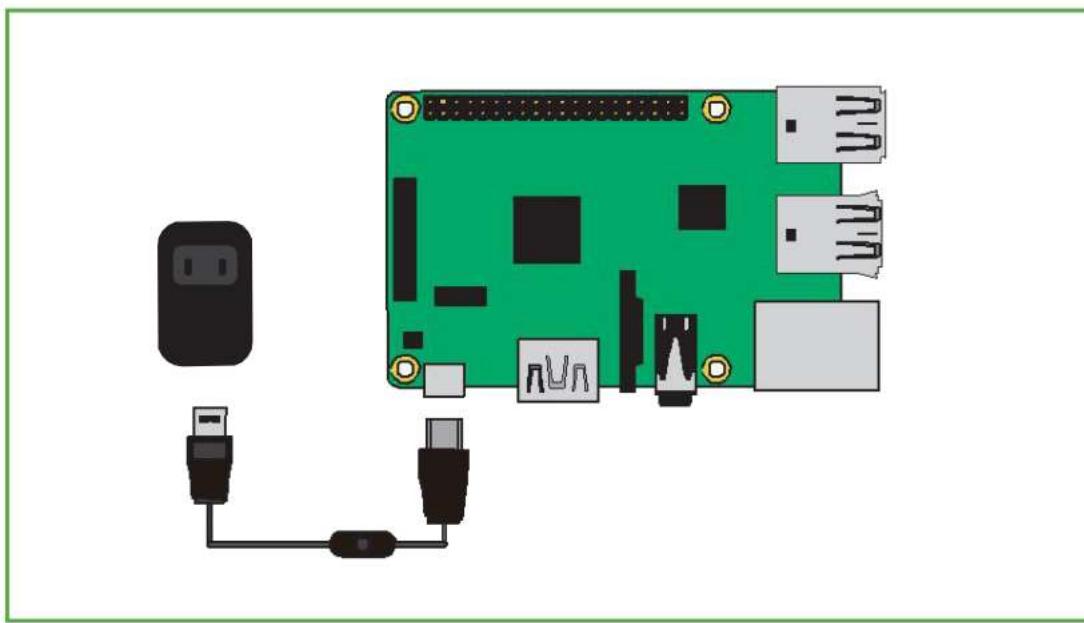
Ensure that the pins are properly aligned when stacking the GrovePi.

Powering up the Raspberry Pi

To power the GrovePi+ and the Raspberry Pi, you can use the micro-USB power port on the Raspberry Pi.

Remember to use a good power adapter capable of supplying 1A at 5V and you should be fine with the power.

If you want to run the GrovePi+ in a standalone configuration, then you should use a USB power bank



Setup the Software on the Raspberry Pi

Next we will install the software on the Raspberry Pi. There are two options for installation:

- You can use our BrickPi Image.
- Use your own image. If you already have your own flavor of linux running on the Raspberry Pi, you can use our bash script to setup for the GrovePi.

Using the BrickPi Image

- Download the Brick Pi Image and install the image on your SD card. [Here is the link to the BrickPi Page with steps to configure the SD card](#). You will need a minimum of 4GB SD Card for this installation.
- Clone the Github repository at an appropriate location in Raspbian

`git clone https://github.com/DexterInd/GrovePi.git`

- Run the bash script in the Scripts folder to configure the Raspbian. [Here is the tutorial for setting up with the Script](#).
- Restart your Raspberry Pi.

Configuring your own image

- Clone the Github repository at an appropriate location

`git clone https://github.com/DexterInd/GrovePi.git`

- Run the bash script in the Scripts folder to configure the Raspbian. [here is the tutorial for setting up with the Script](#).

- Restart the Raspberry Pi and start using the Grove Pi.
- Power on the Raspberry Pi, without the GrovePi attached, and open a terminal (we'll be doing it on SSH, but it the same when using a standard Raspberry Pi setup with a monitor).

```
pi@raspberrypi: ~
login as: pi
pi@169.254.127.191's password:
Linux raspberrypi 3.6.11+ #538 PREEMPT Fri Aug 30 20:42:08 BST 2013 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 30 17:39:35 2013 from 169.254.127.190
pi@raspberrypi ~ $
```

- Change directories go to an appropriate location on your Pi where you want the GrovePi files to be stored (We recommend that you do it on the Desktop because it is easy to access and compatible with all our examples too). Clone the GrovePi git.

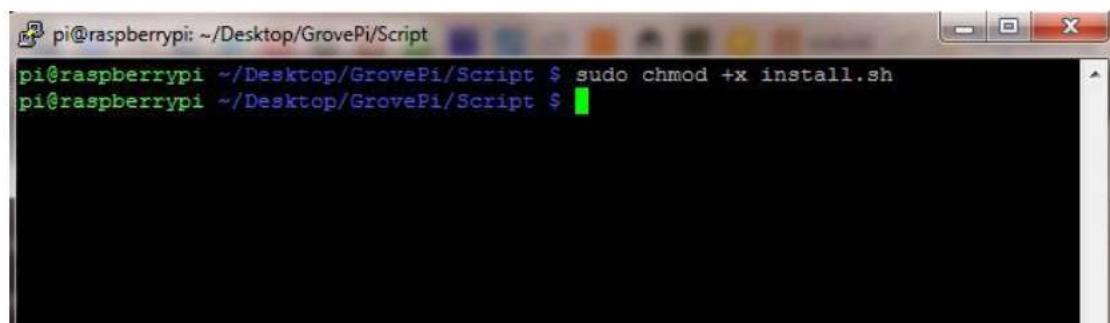
```
git clone https://github.com/DexterInd/GrovePI
```

```
pi@raspberrypi: ~
root@raspberrypi:/home/pi/Desktop# git clone https://github.com/DexterInd/GrovePi.git
Cloning into 'GrovePi'...
remote: Counting objects: 116, done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 116 (delta 38), reused 108 (delta 30)
Receiving objects: 100% (116/116), 157.71 KiB | 32 KiB/s, done.
Resolving deltas: 100% (38/38), done.
root@raspberrypi:/home/pi/Desktop#
```

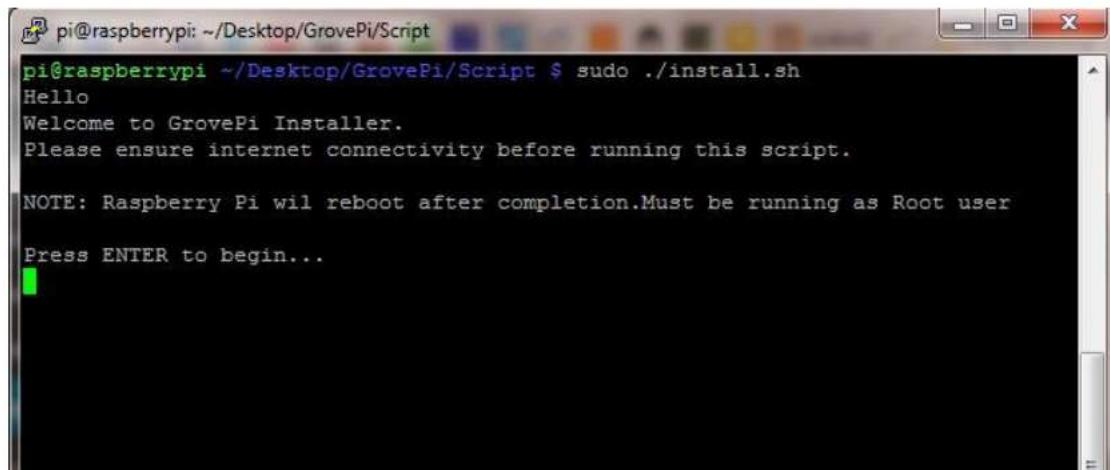
When the repository is done downloading, there should be a new folder called “GrovePi”

- Go to the Scripts folder in the GrovePi folder
cd GrovePi/Script
- Make the install.sh bash script as executable. We do this by modifying the permissions of the script:

```
sudo chmod +x install.sh
```



- Start the script. You must be the root user, so be sure to
sudo ./install.sh



Press “Enter” to start when you are prompted.

- The script will download packages from the internet which are used by the GrovePi. Press “y” when the terminal prompts and asks for permission to start the download.

```
pi@raspberrypi: ~
root@raspberrypi:/home/pi/Desktop/GrovePi/Script# chmod +x install.sh
root@raspberrypi:/home/pi/Desktop/GrovePi/Script# sudo ./install.sh
Hello
Welcome to GrovePi Installer.
Please ensure internet connectivity before running this script.

NOTE: Raspberry Pi will reboot after completion. Must be running as Root user

Press ENTER to begin...

Check for internet connectivity...
=====
Connected

Installing Dependencies
=====
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version.
git set to manually installed.
python-rpi.gpio is already the newest version.
The following extra packages will be installed:
    arduino-core avr-libc avrdude binutils-avr extra-xdg-menus gcc-avr libftdi1
    libjna-java librxtx-java lrzs python-pkg-resources python-setuptools
    python2.6 python2.6-minimal
Suggested packages:
    arduino-mk avrdude-doc task-c-devel gcc-doc gcc-4.2 libjna-java-doc
    python-distribute python-distribute-doc python-wxgtk2.8 python-wxgtk2.6
    python-wxgtk python2.6-doc binfmt-support
Recommended packages:
    python-dev-all
The following NEW packages will be installed:
    arduino arduino-core avr-libc avrdude binutils-avr extra-xdg-menus gcc-avr
    i2c-tools libftdi1 libi2c-dev libjna-java librxtx-java lrzs minicom
    python-pip python-pkg-resources python-serial python-setuptools python-smbus
    python2.6 python2.6-minimal
0 upgraded, 21 newly installed, 0 to remove and 0 not upgraded.
Need to get 29.8 MB of archives.
After this operation, 98.1 MB of additional disk space will be used.
Do you want to continue [Y/n]? █
```

- . The Raspberry Pi will automatically restart when the installation is

```
(Reading database ... 69573 files and directories currently installed.)
Preparing to replace avrdude 5.11.1-1 (using avrdude_5.10-4_armhf.deb) ...
Unpacking replacement avrdude ...
Setting up avrdude (1:5.10-4) ...
Installing new version of config file /etc/avrdude.conf ...
Processing triggers for man-db ...
--2013-12-30 17:32:19-- http://project-downloads.drogon.net/gertboard/setup.sh
Resolving project-downloads.drogon.net (project-downloads.drogon.net)... 195.10.
226.169, 2a00:ce0:2:feed:beef:cafe:0:4
Connecting to project-downloads.drogon.net (project-downloads.drogon.net)|195.10.
.226.169|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1870 (1.8K) [application/x-sh]
Saving to: `setup.sh'

100%[=====] 1,870      --.-K/s   in 0.003s

2013-12-30 17:32:20 (703 KB/s) - `setup.sh' saved [1870/1870]

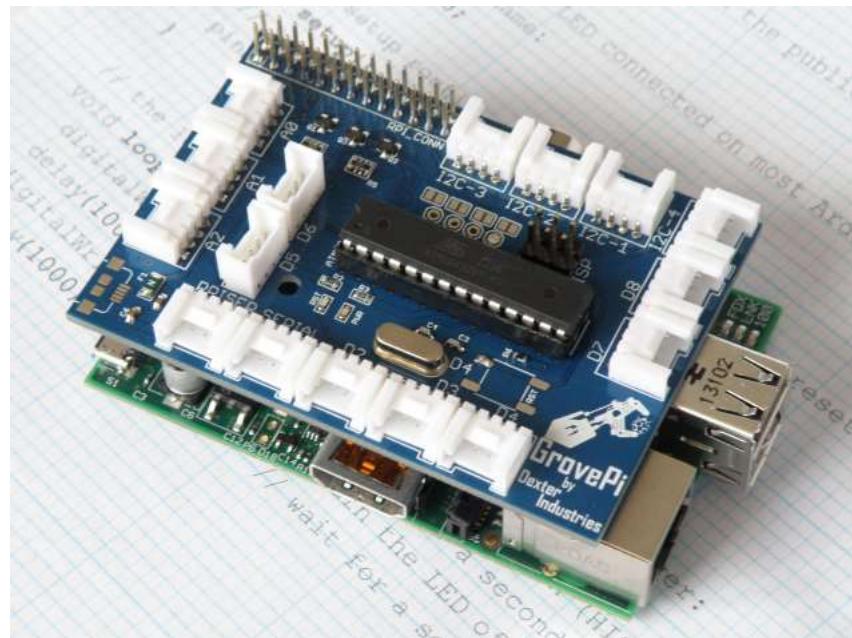
Setting up Raspberry Pi to make it work with the Gertboard
and the ATmega chip on-board with the Arduino IDE.

Checking ...
  Avrdude: OK
  Arduino IDE: OK
Fetching files:
  boards.txt
  programmers.txt
  avrsetup
Replacing/updating files:
  initramfs: OK
  cmdline.txt: OK
  boards.txt: OK
  programmers.txt: OK
All Done.
Check and reboot now to apply changes.

Restarting
3
2
1

Broadcast message from root@raspberrypi (pts/0) (Mon Dec 30 17:32:28 2013):
The system is going down for reboot NOW!
root@raspberrypi:/home/pi/Desktop/GrovePi/Script#
```

- Now when the Raspberry pi is powered down, stack the Grove Pi on top of the Raspberry Pi and power on the Raspberry Pi. A green light should power up on the Grove Pi. (Ensure that the pins are properly connected before powering the Raspberry Pi)



- Now to check that the script was correctly installed. We will check that the Raspberry Pi is able to detect the Grove pi: `runi2cdetect`

```
sudo i2cdetect -y
```

If you have an Original Raspberry Pi (Sold before October 2012) – the I2C is port 0

```
sudo i2cdetect -y
```

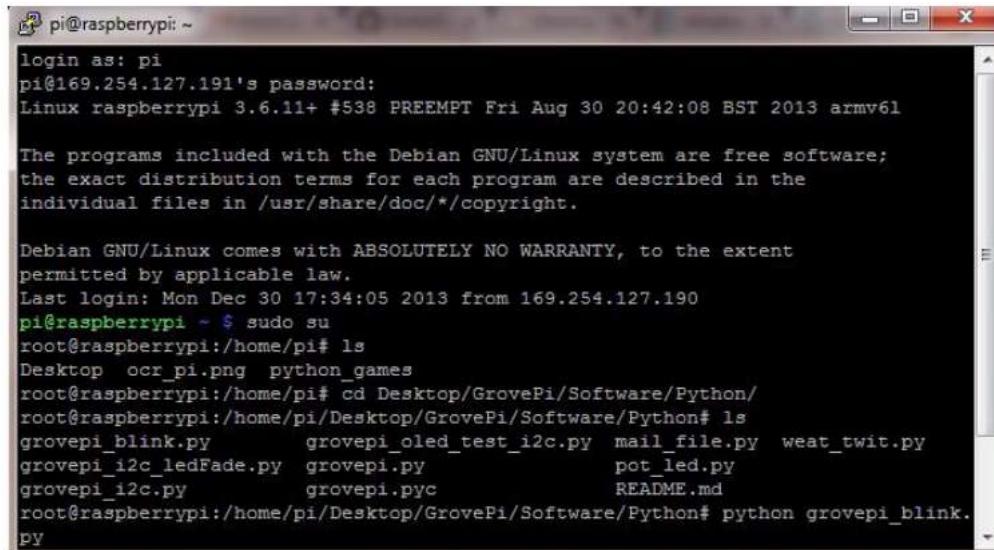
```
pi@raspberrypi: ~/Desktop/GrovePi/Script $ sudo i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- 04 -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- --
pi@raspberrypi: ~/Desktop/GrovePi/Script $
```

If you can see a “04” in the output, this means the Raspberry Pi is able to detect the GroveP

- To test the Grove Pi, connect a Grove LED to port D4 and run the blink example

cd GrovePi/Software/Python

python grovepi_blink.py



```
pi@raspberrypi: ~
login as: pi
pi@169.254.127.191's password:
Linux raspberrypi 3.6.11+ #538 PREEMPT Fri Aug 30 20:42:08 BST 2013 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 30 17:34:05 2013 from 169.254.127.190
pi@raspberrypi ~ $ sudo su
root@raspberrypi:/home/pi# ls
Desktop ocr_pi.png python_games
root@raspberrypi:/home/pi# cd Desktop/GrovePi/Software/Python/
root@raspberrypi:/home/pi/Desktop/GrovePi/Software/Python# ls
grovepi_blink.py      grovepi_oled_test_i2c.py  mail_file.py  weat_twit.py
grovepi_i2c_ledFade.py  grovepi.py            pot_led.py
grovepi_i2c.py        grovepi.pyc          README.md
root@raspberrypi:/home/pi/Desktop/GrovePi/Software/Python# python grovepi_blink.py
```

If everything is installed correctly, the LED should start

Unit 5: Interface Grove Pi+ Sensors to Raspberry Pi

Learning Outcomes:

- Interface grovePi+ with Raspberry Pi
- Able to install grovePi+ sensors to Raspberry Pi
- Able to interface various sensors and actuators

5.1 Practical: Interface Light Sensor with raspberry Pi

Hardware

Step 1: Prepare the below stuffs:

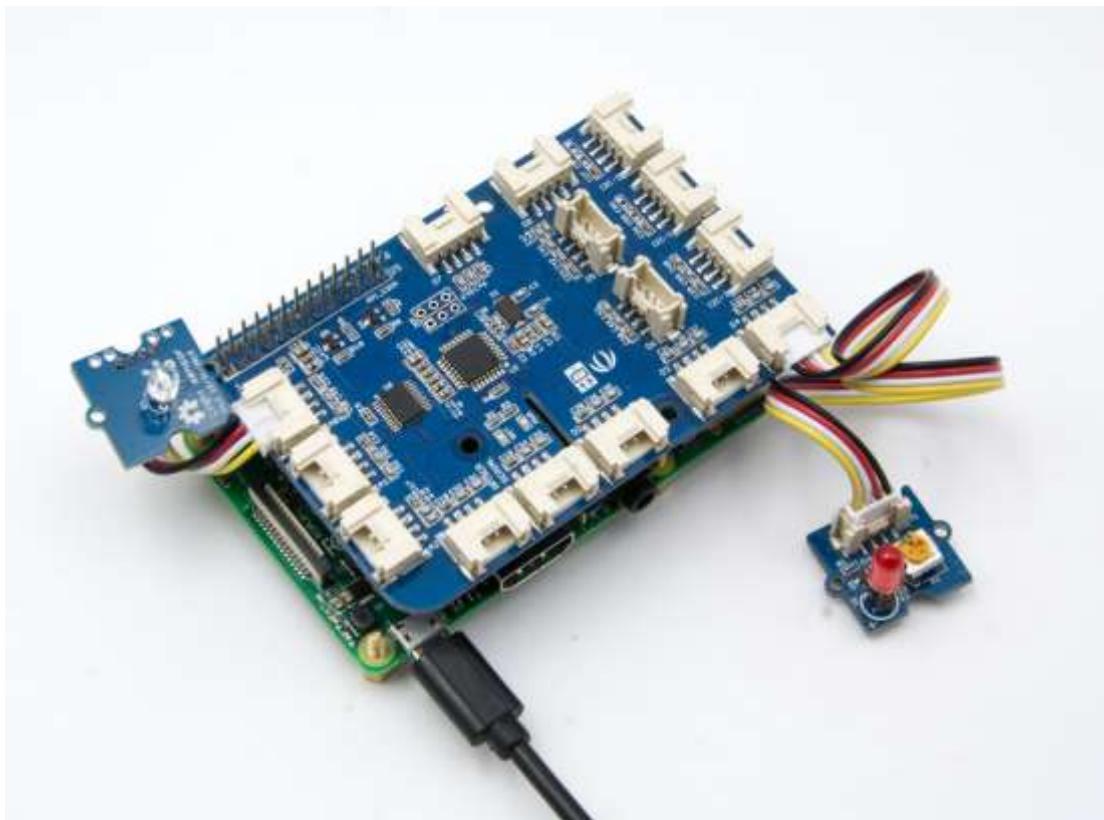


Step 2: Plug the GrovePi_Plus into Raspberry.

Step 3: Connect Grove-light sensor to A0 port of GrovePi_Plus.

Step 4: Connect Grove-Red Led to D4 port of GrovePi_Plus.

Step 5: Connect the Raspberry to PC through USB cable.



Software

Step 1: Follow [Setting Software](#) to configure the development environment.

Step 2: Git clone the Github repository.

```
cd ~  
git clone https://github.com/DexterInd/GrovePi.git
```

Step 3: Execute below commands.

```
cd ~/GrovePi/Software/Python  
python3 grove_light_sensor.py
```

Here is the grove_light_sensor.py code.

```
import time  
import grovepi
```

```
# Connect the Grove Light Sensor to analog port A0  
# SIG,NC,VCC,GND  
light_sensor = 0
```

```
# Connect the LED to digital port D4  
# SIG,NC,VCC,GND  
led = 4
```

```

# Turn on LED once sensor exceeds threshold resistance
threshold = 10

grovepi.pinMode(light_sensor,"INPUT")
grovepi.pinMode(led,"OUTPUT")

while True:
    try:
        # Get sensor value
        sensor_value = grovepi.analogRead(light_sensor)

        # Calculate resistance of sensor in K
        resistance = (float)(1023 - sensor_value) * 10 / sensor_value

        if resistance > threshold:
            # Send HIGH to switch on LED
            grovepi.digitalWrite(led,1)
        else:
            # Send LOW to switch off LED
            grovepi.digitalWrite(led,0)

        print("sensor_value = %d resistance = %.2f" %(sensor_value, resistance))
        time.sleep(.5)

    except IOError:
        print ("Error")

```

Step 4: The led will turn on when the light sensor gets covered.

```

pi@raspberrypi:~/GrovePi/Software/Python $ python3 grove_light_sensor.py
sensor_value = 754 resistance = 3.57
sensor_value = 754 resistance = 3.57
sensor_value = 752 resistance = 3.60
sensor_value = 752 resistance = 3.60
sensor_value = 752 resistance = 3.60
sensor_value = 313 resistance = 22.68
sensor_value = 155 resistance = 56.00
sensor_value = 753 resistance = 3.59

```

5.2 Practical: Interface Sound Sensor with raspberry Pi

Hardware

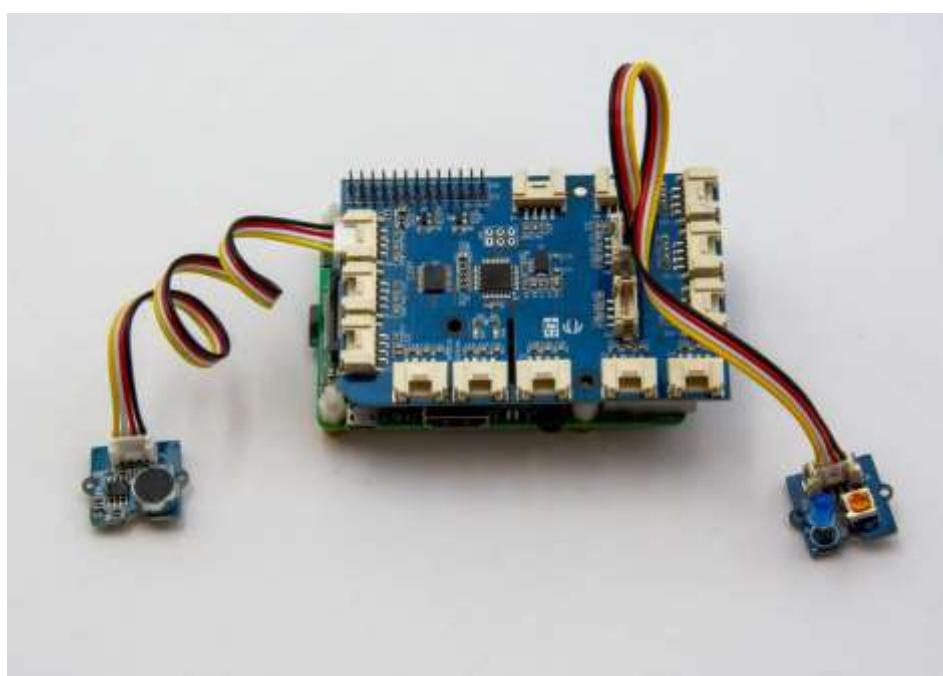
Step 1: Prepare the below stuffs:



Step 2: Plug the GrovePi_Plus into Raspberry.

Step 3: Connect Grove-Sound Sensor to A0 port of GrovePi_Plus , and connect Grove-Blue LED to D5 port of GrovePi_Plus

Step 4: Connect the Raspberry to PC through USB cable.



Software

Step 1: Follow [Setting Software](#) to configure the development environment.

Step 2: Follow [Updating the Firmware](#) to update the newest firmware of GrovePi.

Step 3: Git clone the Github repository.

```
cd ~  
git clone https://github.com/DexterInd/GrovePi.git
```

Step 4: Navigate to the demos' directory:

```
cd yourpath/GrovePi/Software/Python/
```

Here is the grove_sound_sensor.py code.

```
import time  
  
import grovepi  
  
  
# Connect the Grove Sound Sensor to analog port A0  
# SIG,NC,VCC,GND  
sound_sensor = 0  
  
  
# Connect the Grove LED to digital port D5  
# SIG,NC,VCC,GND  
led = 5  
  
  
grovepi.pinMode(sound_sensor,"INPUT")  
grovepi.pinMode(led,"OUTPUT")  
  
  
# The threshold to turn the led on 400.00 * 5 / 1024 = 1.95v  
threshold_value = 400  
  
  
while True:
```

try:

```
# Read the sound level
sensor_value = grovepi.analogRead(sound_sensor)
```

```
# If loud, illuminate LED, otherwise dim
```

```
if sensor_value > threshold_value:
```

```
    grovepi.digitalWrite(led,1)
```

```
else:
```

```
    grovepi.digitalWrite(led,0)
```

```
print("sensor_value = %d" %sensor_value)
```

```
time.sleep(.5)
```

```
except IOError:
```

```
    print ("Error")
```

Step 5: Run the demo.

```
sudo python3 grove_sound_sensor.py
```

5.3 Practical: Interface Grove LCD with Raspberry Pi

Hardware setup

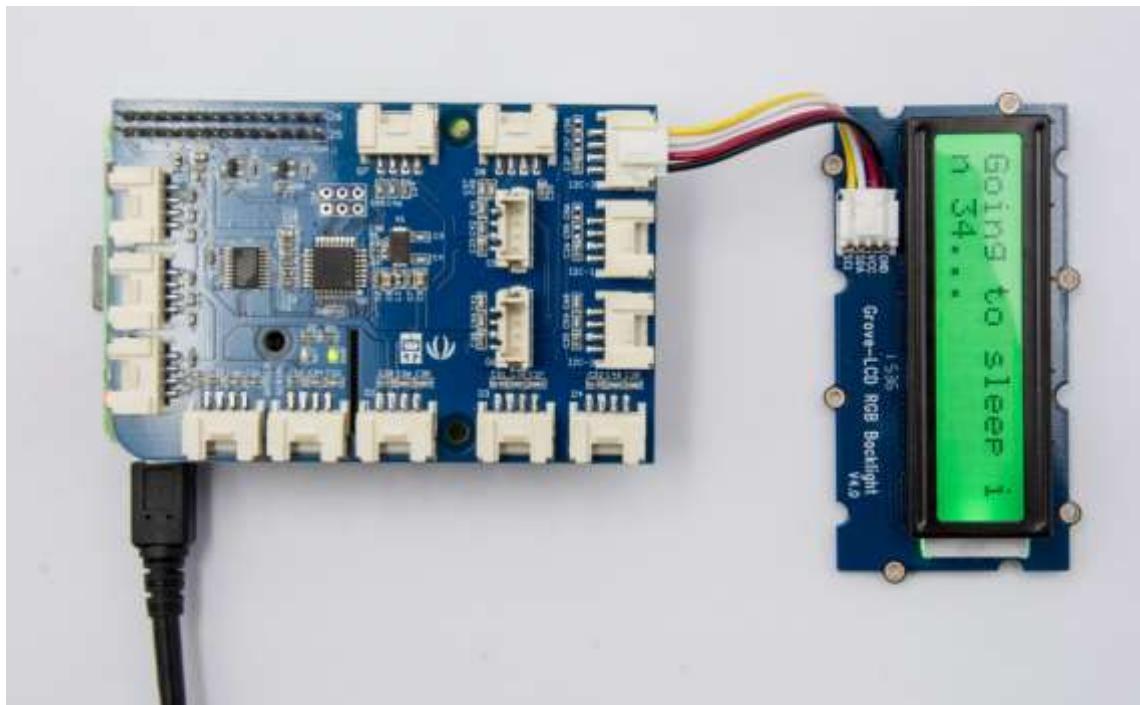
Step 1: Prepare the below stuffs:



Step 2: Plug the GrovePi_Plus into Raspberry.

Step 3: Connect Grove-LCD RGB Backlight to I2C port of GrovePi_Plus.

Step 4: Connect the Raspberry to PC through USB cable.



Software setup

Step 1: Follow Setting Software to configure the development environment.

Step 2: Git clone the Github repository.

```
cd ~  
git clone https://github.com/DexterInd/GrovePi.git
```

Step 3: Execute below commands to use the Grove-LCD RGB Backlight to display.

```
cd ~/GrovePi/Software/Python/grove_rgb_lcd  
python3 grove_rgb_lcd.py
```

Here is the grove_rgb_lcd.py code.

```
import time,sys
```

```
if sys.platform == 'uwp':
```

```
    import winrt_smbus as smbus
```

```
bus = smbus.SMBus(1)

else:
    import smbus
    import RPi.GPIO as GPIO
    rev = GPIO.RPI_REVISION
    if rev == 2 or rev == 3:
        bus = smbus.SMBus(1)
    else:
        bus = smbus.SMBus(0)

# this device has two I2C addresses
DISPLAY_RGB_ADDR = 0x62
DISPLAY_TEXT_ADDR = 0x3e

# set backlight to (R,G,B) (values from 0..255 for each)
def setRGB(r,g,b):
    bus.write_byte_data(DISPLAY_RGB_ADDR,0,0)
    bus.write_byte_data(DISPLAY_RGB_ADDR,1,0)
    bus.write_byte_data(DISPLAY_RGB_ADDR,0x08,0xaa)
    bus.write_byte_data(DISPLAY_RGB_ADDR,4,r)
    bus.write_byte_data(DISPLAY_RGB_ADDR,3,g)
    bus.write_byte_data(DISPLAY_RGB_ADDR,2,b)

# send command to display (no need for external use)
def textCommand(cmd):
    bus.write_byte_data(DISPLAY_TEXT_ADDR,0x80,cmd)

# set display text \n for second line(or auto wrap)
def setText(text):
```

```
textCommand(0x01) # clear display
time.sleep(.05)
textCommand(0x08 | 0x04) # display on, no cursor
textCommand(0x28) # 2 lines
time.sleep(.05)
count = 0
row = 0
for c in text:
    if c == '\n' or count == 16:
        count = 0
        row += 1
        if row == 2:
            break
        textCommand(0xc0)
    if c == '\n':
        continue
    count += 1
    bus.write_byte_data(DISPLAY_TEXT_ADDR,0x40,ord(c))
```

#Update the display without erasing the display

```
def setText_norefresh(text):
    textCommand(0x02) # return home
    time.sleep(.05)
    textCommand(0x08 | 0x04) # display on, no cursor
    textCommand(0x28) # 2 lines
    time.sleep(.05)
    count = 0
    row = 0
    while len(text) < 32: #clears the rest of the screen
```

```

text += ' '
for c in text:
    if c == '\n' or count == 16:
        count = 0
        row += 1
        if row == 2:
            break
        textCommand(0xc0)
    if c == '\n':
        continue
    count += 1
    bus.write_byte_data(DISPLAY_TEXT_ADDR,0x40,ord(c))

# example code
if __name__=="__main__":
    setText("Hello world\nThis is an LCD test")
    setRGB(0,128,64)
    time.sleep(2)
    for c in range(0,255):
        setText_norefresh("Going to sleep in {}...".format(str(c)))
        setRGB(c,255-c,0)
        time.sleep(0.1)
    setRGB(0,255,0)
    setText("Bye bye, this should wrap onto next line")

```

Step 4: We will see the Grove-LCD RGB Backlight display as Going to sleep in 1...

5.4 Practical: Interface Grove Button with Raspberry Pi

Hardware

Step 1: Prepare the below stuffs:



Step 2: Plug the GrovePi_Plus into Raspberry.

Step 3: Connect Grove-Button to D3 port of GrovePi_Plus.

Step 4: Connect the Raspberry to PC through USB cable.



Software

Step 1: Follow [Setting Software](#) to configure the development environment.

Step 2: Git clone the Github repository.

`cd ~`

`git clone https://github.com/DexterInd/GrovePi.git`

Step 3: Execute below commands.

```
cd  
~/GrovePi/Software/Python  
python3 grove_button.py
```

Here is the grove_button.py code.

```
import time  
  
import grovepi  
  
# Connect the Grove Button to digital port D3  
# SIG,NC,VCC,GND  
button = 3  
  
grovepi.pinMode(button,"INPUT")  
  
while True:  
    try:  
        print(grovepi.digitalRead(button))  
        time.sleep(.5)  
  
    except IOError:  
        print ("Error")
```

Step 4: We will see the button on and off.

```
pi@raspberrypi:~/GrovePi/Software/Python $ python3 grove_button.py  
0  
1  
1  
1  
1  
0  
0
```

5.5 Practical: Interface Grove Temperature and humidity sensor (DHT22) with Raspberry Pi

Hardware

Step 1: Prepare below stuff

Raspberry pi	GrovePi_Plus	Grove-Rotary Angle Sensor	Grove-LED
			
Get ONE Now			

Step 2: Plug the GrovePi_Plus into Raspberry.

Step 3: Connect Grove - Temperature & Humidity Sensor Pro to D4 port of GrovePi+

Step 4: Connect the Raspberry to PC via USB cable.



Software

If this is the first time you use GrovePi, please do this part step by step. If you are an old friend with GrovePi, you can skip **Step1** and **Step2**.

Step 1: Setting Up The Software. In the command line, type the following commands:

```
sudo curl -kL dexterindustries.com/update_grovepi | bash
sudo reboot
cd/home/pi/Desktop
git clone https://github.com/DexterInd/GrovePi.git
```

Step 2: Follow Updating the Firmware to update the newest firmware of GrovePi.

Step 3: Configure Parameter

```
cd /home/pi/Desktop/GrovePi/Software/Python/
sudo nano grove_dht_pro.py
```

Change the default parameter [temp,humidity] =
grovepi.dht(sensor,blue) into [temp,humidity] =
grovepi.dht(sensor,white). Then the code should be like:

```
import grovepi
import math
# Connect the Grove Temperature & Humidity Sensor Pro to digital port D4
# This example uses the blue colored sensor.
# SIG,NC,VCC,GND
sensor = 4 # The Sensor goes on digital port 4.

# temp_humidity_sensor_type
# Grove Base Kit comes with the blue sensor.
blue = 0 # The Blue colored sensor.
white = 1 # The White colored sensor.

while True:
    try:
        # This example uses the blue colored sensor.
        # The first parameter is the port, the second parameter is the type of sensor.
        [temp,humidity] = grovepi.dht(sensor,white)
        if math.isnan(temp) == False and math.isnan(humidity) == False:
            print("temp = %.02f C humidity =%.02f%%"%(temp, humidity))

    except IOError:
        print ("Error")
```

Then tap Ctrl+X to quit nano. Tap Y to save the change.

Step 4: Run the following command to get the result.

```
sudo python3 grove_dht_pro.py
```

Result

```
pi@raspberrypi:~/GrovePi/Software/Python $ sudo python3 grove_dht_pro.py
temp = 22.90 C humidity =42.30%
```

5.6 Practical: Interface Grove Relay Switch with Raspberry Pi



The Grove-Relay module is a digital normally-open switch. Through it, you can control circuit of high voltage with low voltage, say 5V on the controller. There is an indicator LED on the board, which will light up when the controlled terminals get closed.

Hardware

Step 1: Prepare the below Hardware

Raspberry pi

GrovePi_Plus

Grove-Button

Grove-Relay

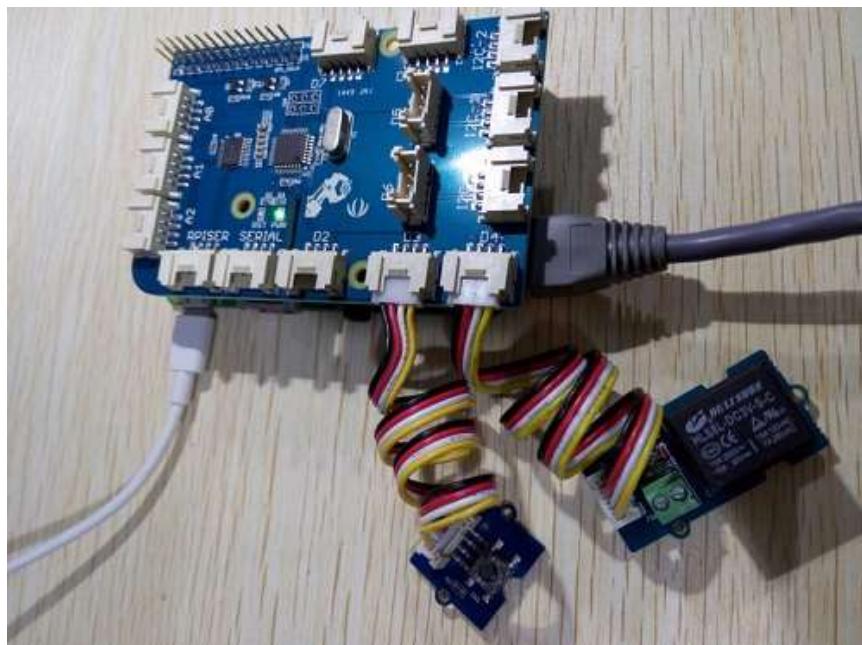


Step 2: Plug the GrovePi_Plus into Raspberry.

Step 3: Connect the Grove-Relay to D4 port of GrovePi_Plus.

Step 4: Connect the Grove-Button to D3 port of GrovePi_Plus.

Step 5: Connect the Raspberry to PC via USB cable.



Software

Step 1: Setting Up The Software. In the command line, type the following commands:

```
sudo curl -kL dexterindustries.com/update_grovepi | bash
```

```
sudo reboot
```

```
cd /home/pi/Desktop
```

```
git clone https://github.com/DexterInd/GrovePi.git
```

Step 2: Follow Updating the Firmware to update the latest firmware of GrovePi.

Step 3: Run the following command to get the result.

```
cd /home/pi/Desktop/GrovePi/Software/Python/
sudo python3 grove_switch_relay.py
```

If you want to check the code, you can use the following command:

```
sudo nano grove_switch_relay.py
```

Code

```
# Raspberry Pi + Grove Switch + Grove
Relay
```

```
import time
import grovepi
# Connect the Grove Switch to digital port D3
# SIG,NC,VCC,GND

switch = 3
# Connect the Grove Relay to digital port D4
# SIG,NC,VCC,GND

relay = 4
grovepi.pinMode(switch,"INPUT")
grovepi.pinMode(relay,"OUTPUT")
while True:
    try:
        if grovepi.digitalRead(switch):
            grovepi.digitalWrite(relay,1)
        else:
            grovepi.digitalWrite(relay,0)
            time.sleep(.05)
    except KeyboardInterrupt:
        grovepi.digitalWrite(relay,0)
        break
    except IOError:
        print "Error"
```

5.7 Practical: Interface Grove Ultrasonic sensor with Raspberry Pi



This Grove - Ultrasonic ranger is a non-contact distance measurement module which works at 40KHz. When we provide a pulse trigger signal with more than 10uS through signal pin, the Grove_Ultrasonic_Ranger will issue 8 cycles of 40kHz cycle level and detect the echo. The pulse width of the echo signal is proportional to the measured distance. Here is the formula: Distance = echo signal high time * Sound speed (340M/S)/2. Grove_Ultrasonic_Ranger's trig and echo signal share 1 SIG pin.

Hardware

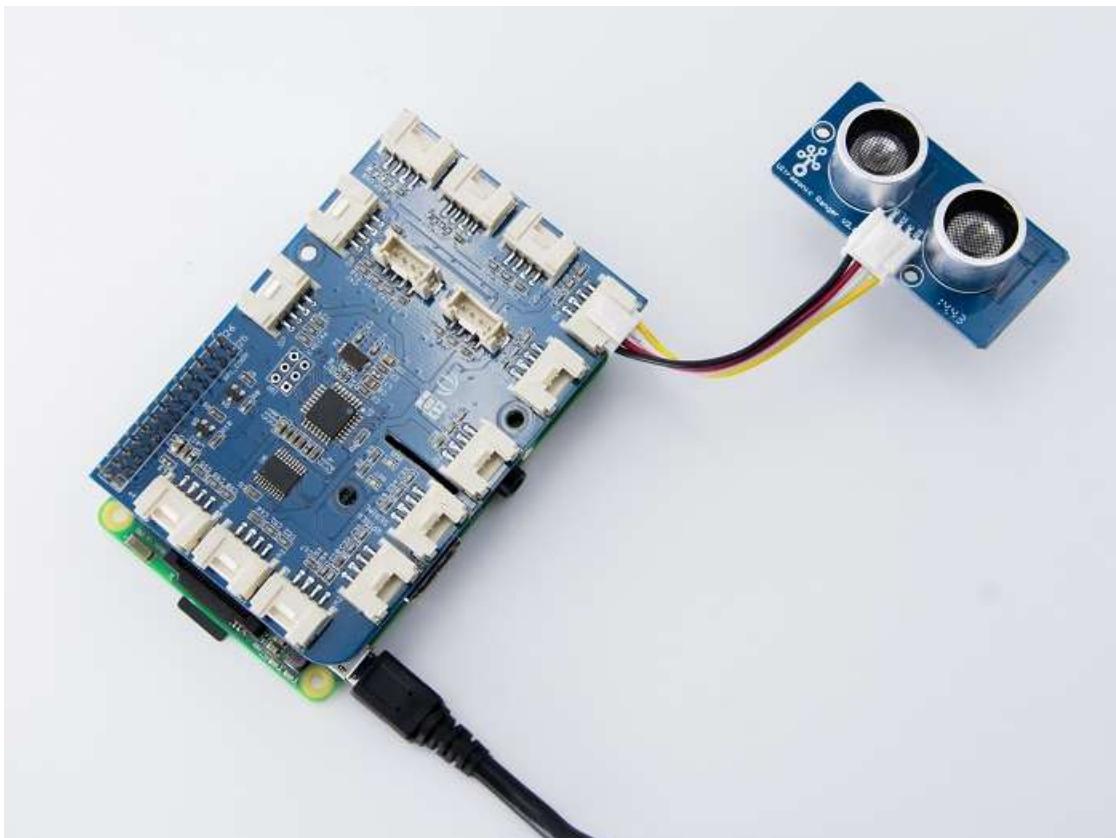
Step 1: Prepare the below Hardware



Step 2: Plug the GrovePi_Plus into Raspberry.

Step 3: Connect Grove-Ultrasonic ranger to **D4** port of GrovePi_Plus.

Step 4: Connect the Raspberry to PC through USB cable.



Software

Step 1: Follow [Setting Software](#) to configure the development environment.

Step 2: Git clone the Github repository.

```
cd ~
```

```
git clone https://github.com/DexterInd/GrovePi.git
```

Step 3: Execute below commands to use the ultrasonic_ranger to measure the distance.

```
cd ~/GrovePi/Software/Python
```

```
python3 grove_ultrasonic.py
```

Here is the grove_ultrasonic.py code.

```
# GrovePi + Grove Ultrasonic Ranger
```

```
from grovepi import *
```

```
# Connect the Grove Ultrasonic Ranger to digital port D4
```

```
# SIG,NC,VCC,GND

ultrasonic_ranger = 4

while True:
    try:
        # Read distance value from Ultrasonic
        print ultrasonicRead(ultrasonic_ranger)

    except TypeError:
        print "Error"
    except IOError:
        print "Error"
```

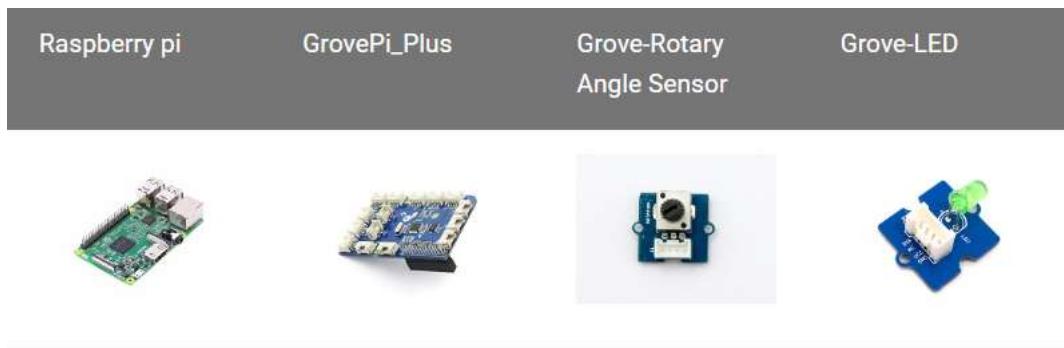
Step 4: We will see the distance display on terminal as below.

```
pi@raspberrypi:~/GrovePi/Software/Python $ python3 grove_ultrasonic.py
9
9
9
9
9
9
9
9
9
9
9
9
9
9
```

5.8 Practical: Interface Grove Rotary Angle sensor with Raspberry Pi

Hardware setup

Step 1: Prepare below stuff

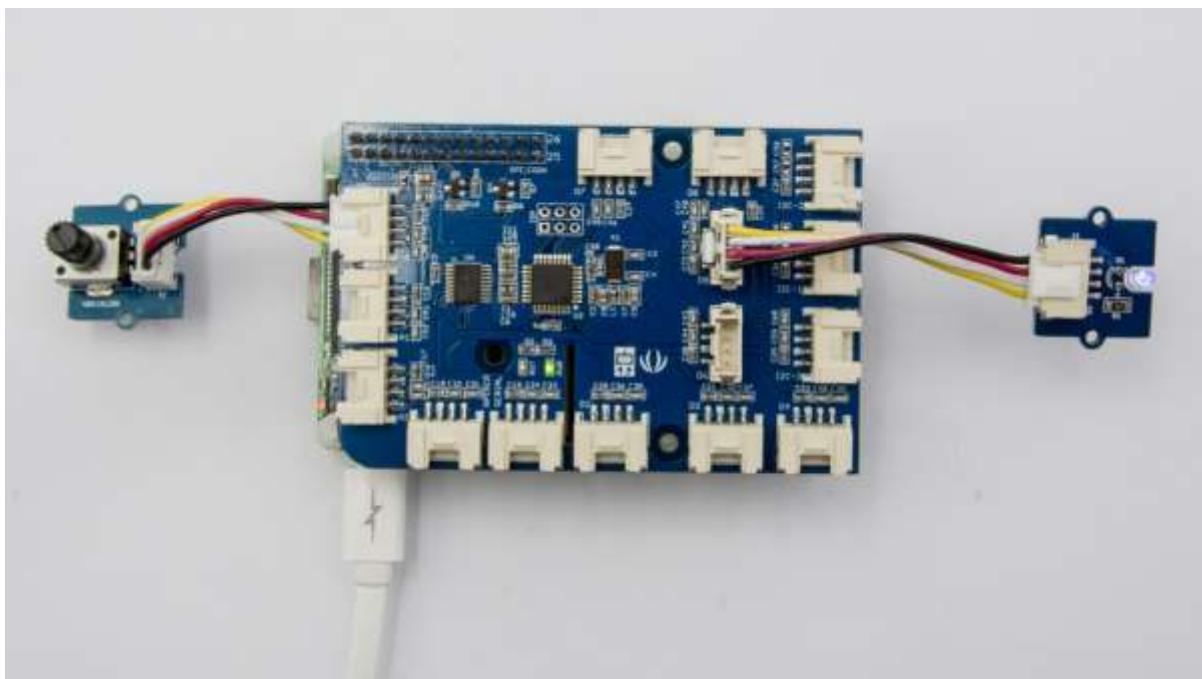


Step 2: Plug the GrovePi_Plus into Raspberry.

Step 3: Connect Grove-Rotary Angle Sensor to A0 port of GrovePi_Plus.

Step 4: Connect Grove-LED to D5 port of GrovePi_Plus.

Step 5: Connect the Raspberry to PC through USB cable.



Software setup

Step 1: Follow Setting Software to configure the development environment.

Step 2: Git clone the Github repository.

cd ~

git clone https://github.com/DexterInd/GrovePi.git

Step 3: Execute below commands to monitor the loudness.

cd ~/GrovePi/Software/Python

python3 grove_rotary_angle_sensor.py

Here is the grove_rotary_angle_sensor.py code.

Code:

```
import time
import grovepi
```

```
# Connect the Grove Rotary Angle Sensor to analog port A0
# SIG,NC,VCC,GND
potentiometer = 0
```

```
# Connect the LED to digital port D5
# SIG,NC,VCC,GND
led = 5
```

```
grovepi.pinMode(potentiometer,"INPUT")
grovepi.pinMode(led,"OUTPUT")
time.sleep(1)
```

```
# Reference voltage of ADC is 5v
adc_ref = 5
```

```
# Vcc of the grove interface is normally 5v
grove_vcc = 5
```

```
# Full value of the rotary angle is 300 degrees, as per it's specs (0 to 300)
full_angle = 300
```

```
while True:
```

```
    try:
```

```
        # Read sensor value from potentiometer
```

```
        sensor_value = grovepi.analogRead(potentiometer)
```

```

# Calculate voltage
voltage = round((float)(sensor_value) * adc_ref / 1023, 2)

# Calculate rotation in degrees (0 to 300)
degrees = round((voltage * full_angle) / grove_vcc, 2)

# Calculate LED brightness (0 to 255) from degrees (0 to 300)
brightness = int(degrees / full_angle * 255)

# Give PWM output to LED
grovepi.analogWrite(led,brightness)

print("sensor_value = %d voltage = %.2f degrees = %.1f brightness = %d"
%(sensor_value, voltage, degrees, brightness))
except KeyboardInterrupt:
    grovepi.analogWrite(led,0)
    break
except IOError:
    print ("Error")

```

Step 4: Adjust Grove-Rotary Angle Sensor and we will see the Grove-LED changes the brightness.

5.9 Practical: Creating GUI interfaces to communicate with sensor and displays

Lets create a GUI interface using Python's prominent Library Tkinter. Here we are going to create a simple window wit button widget to control Led's. Click command of button widget transfer the operation to execution of hardware controlling python files.

```

import time
from grovepi import *

led = 7
pinMode(led,"OUTPUT")
digitalWrite(led,1)

```

```
#!/usr/bin/python

import tkinter as Tkinter
import time
from subprocess import Popen

Freq = 2500
Dur = 150

top = Tkinter.Tk()
top.title('GUI_Control_RPI')
top.geometry('300x300') # Size 200, 200

def start():
    import os
    # os.system("python test.py")
    Popen(["python3", "test1.py"])

def stop():
    import os
    # os.system("python test.py")
    Popen(["python3", "test2.py"])

def close():
    print ("Stop")
    top.destroy()

startButton = Tkinter.Button(top, height=2, width=20, text ="Start",
command = start)
stopButton = Tkinter.Button(top, height=2, width=20, text ="Stop",
```

```
command = stop)

closeButton = Tkinter.Button(top, height=2, width=20, text ="Close",
command = close)

startButton.pack()
stopButton.pack()
closeButton.pack()

top.mainloop()

import time

from grovepi import *

led = 7

pinMode(led,"OUTPUT")

digitalWrite(led,0)
```

Self-practice exercises

Practical: Real life use case and solve problem using IoT

Module III

Deep Learning, Computer Vision & Edge Computing with OpenVINO toolkit

Unit 1: Deep Learning

Learning Outcomes:

- Understand the concept and features of Deep Learning
- Able to solve problems using deep learning and neural network
- Understand the concept of forward/backward Propagation

1.1 What is Deep Learning?

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Deep learning neural networks, or artificial neural networks, attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.

1.2 Architecture

Deep Neural Networks

It is a neural network that incorporates the complexity of a certain level, which means several numbers of hidden layers are encompassed in between the input and output layers. They are highly proficient in modelling and processing non-linear associations.

Deep Belief Networks

A deep belief network is a class of Deep Neural Network that comprises of multi-layer belief networks.

Steps to perform DBN:

- With the help of the Contrastive Divergence algorithm, a layer of features is learned from perceptible units.
- Next, the formerly trained features are treated as visible units, which perform learning of features.
- Lastly, when the learning of the final hidden layer is accomplished, then the whole DBN is trained.

Recurrent Neural Networks

It permits parallel as well as sequential computation, and it is exactly similar to that of the human brain (large feedback network of connected neurons). Since they are capable enough to reminisce all of the imperative things related to the input they have received, so they are more precise.

1.3 Deep learning vs. Machine learning

Parameter	Machine Learning	Deep Learning
Data Dependency	Although machine learning depends on the huge amount of data, it can work with a smaller amount of data.	Deep Learning algorithms highly depend on a large amount of data, so we need to feed a large amount of data for good performance.
Execution time	Machine learning algorithm takes less time to train the model than deep learning, but it takes a long-time duration to test the model.	Deep Learning takes a long execution time to train the model, but less time to test the model.

Hardware Dependencies	Since machine learning models do not need much amount of data, so they can work on low-end machines.	The deep learning model needs a huge amount of data to work efficiently, so they need GPU's and hence the high-end machine.
Feature Engineering	Machine learning models need a step of feature extraction by the expert, and then it proceeds further.	Deep learning is the enhanced version of machine learning, so it does not need to develop the feature extractor for each problem; instead, it tries to learn high-level features from the data on its own.
Problem-solving approach	To solve a given problem, the traditional ML model breaks the problem in sub-parts, and after solving each part, produces the result.	The problem-solving approach of a deep learning model is different from the traditional ML model, as it takes input for a given problem, and produce the end result. Hence it follows the end-to-end approach.
Interpretation of result	The interpretation of the result for a given problem is easy. As when we work with machine learning, we can interpret the result easily, it means why this result occur, what was the process.	The interpretation of the result for a given problem is very difficult. As when we work with the deep learning model, we may get a better result for a given problem than the machine learning model, but we cannot find why this particular outcome occurred, and the reasoning.
Type of data	Machine learning models mostly require data in a structured form.	Deep Learning models can work with structured and unstructured data both as they rely on the layers of the Artificial neural network.
Suitable for	Machine learning models are suitable for solving simple or bit-complex problems.	Deep learning models are suitable for solving complex problems.

1.4 Concept of Neural Networks

Artificial neural networks (ANNs), usually simply called neural networks (NNs), are computing systems vaguely inspired by the biological neural networks that constitute animal brains.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

Neural networks learn (or are trained) by processing examples, each of which contains a known "input" and "result," forming probability-weighted associations between the two, which are stored within the data structure of the net itself. The training of a neural network from a given example is usually conducted by determining the difference between the processed output of the network (often a prediction) and a target output. This is the error. The network then adjusts its weighted associations according to a learning rule and using the error value. Successive adjustments will cause the neural network to produce output which is increasingly similar to the target output. After a sufficient number of these adjustments the training can be terminated based upon certain criteria. This is known as supervised learning.

Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example,

that the cats have fur, tails, whiskers, and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process.

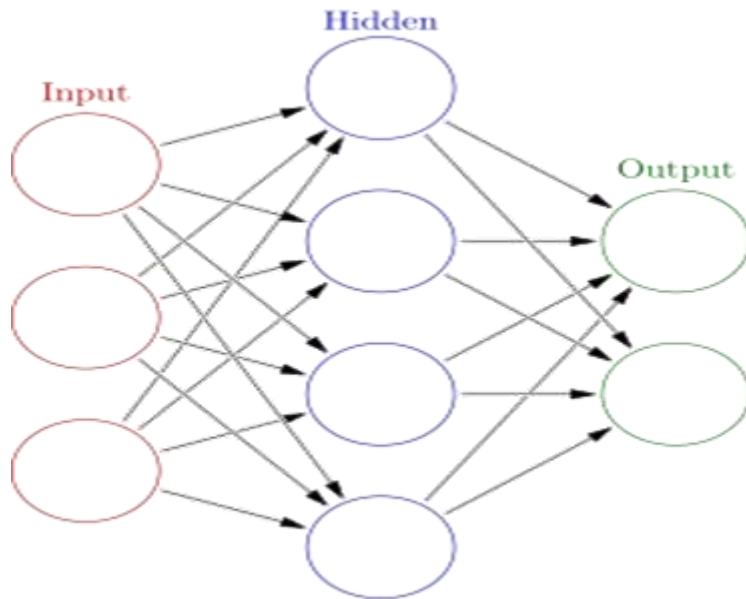


Fig: Neural Network

Reference - Glosser.ca, CC BY-SA 3.0 <<https://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons

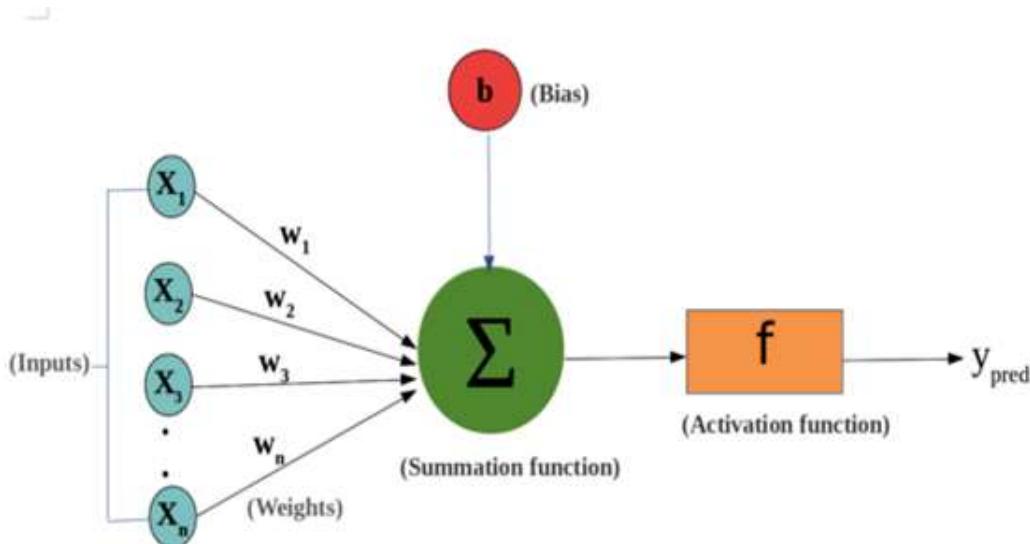


Fig: Basic building blocks of Neural Networks

Reference - <https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f>

Let us understand the core part of artificial neural networks which is Neuron.

Neurons

ANNs are composed of artificial neurons which are conceptually derived from biological neurons. Each artificial neuron has inputs and produces a single output which can be sent to multiple other neurons. The inputs can be the feature values of a sample of external data, such as images or documents, or they can be the outputs of other neurons. The outputs of the final output neurons of the neural net accomplish the task, such as recognizing an object in an image.

To find the output of the neuron, first we take the weighted sum of all the inputs, weighted by the weights of the connections from the inputs to the neuron. We add a bias term to this sum. This weighted sum is sometimes called the activation. This weighted sum is then passed through a (usually nonlinear) activation function to produce the output. The initial inputs are external data, such as images and documents. The ultimate outputs accomplish the task, such as recognizing an object in an image.

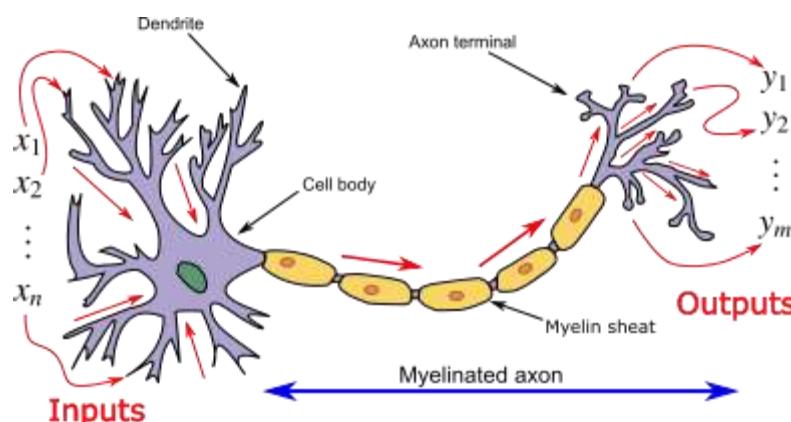


Fig: Neuron

Reference - Egm4313.s12 at English Wikipedia, CC BY-SA 3.0 <<https://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons

Activation Functions

The activation function of a node defines the output of that node given an input or set of inputs.

What is Activation Function?

It's just a thing function that you use to get the output of node. It is also known as Transfer Function.

Why we use Activation functions with Neural Networks?

It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending upon the function).

The Activation Functions can be basically divided into 2 types-

3. Linear Activation Function
4. Non-linear Activation Functions

FYI: The Cheat sheet is given below.

Linear or Identity Activation Function

As you can see the function is a line or linear. Therefore, the output of the functions will not be confined between any range.

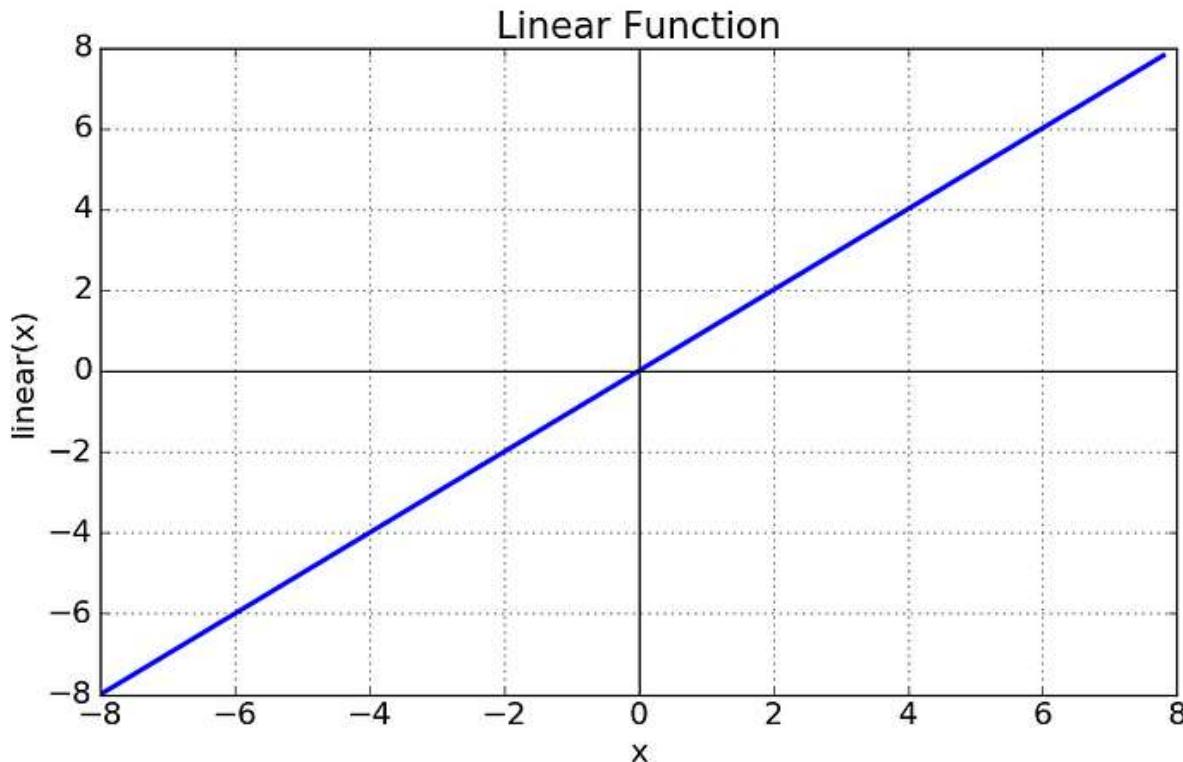


Fig: Linear Activation Function

Equation : $f(x) = x$

Range : (-infinity to infinity)

It doesn't help with the complexity or various parameters of usual data that is fed to the neural networks.

Non-linear Activation Function

The Nonlinear Activation Functions are the most used activation functions. Nonlinearity helps to makes the graph look something like this

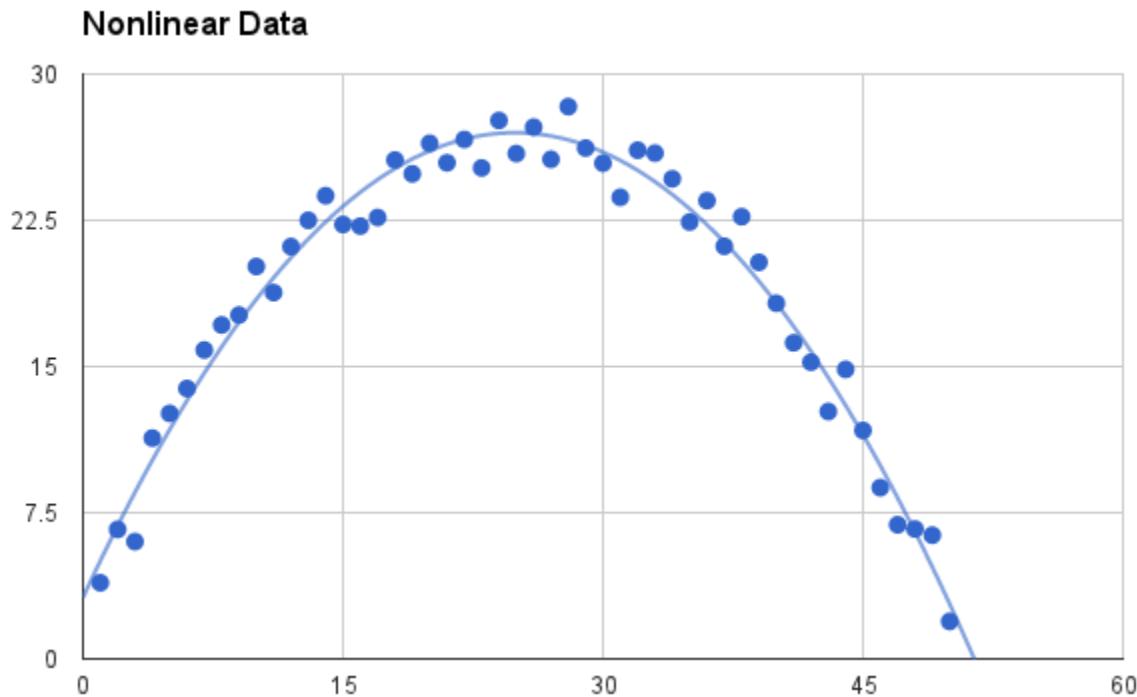


Fig: Non-linear Activation Function

It makes it easy for the model to generalize or adapt with variety of data and to differentiate between the output.

The main terminologies needed to understand for nonlinear functions are:

Derivative or Differential: Change in y-axis w.r.t. change in x-axis. It is also known as slope.

Monotonic function: A function which is either entirely non-increasing or non-decreasing.

The Nonlinear Activation Functions are mainly divided on the basis of their range or curves-

1. Sigmoid or Logistic Activation Function

The Sigmoid Function curve looks like a S-shape.

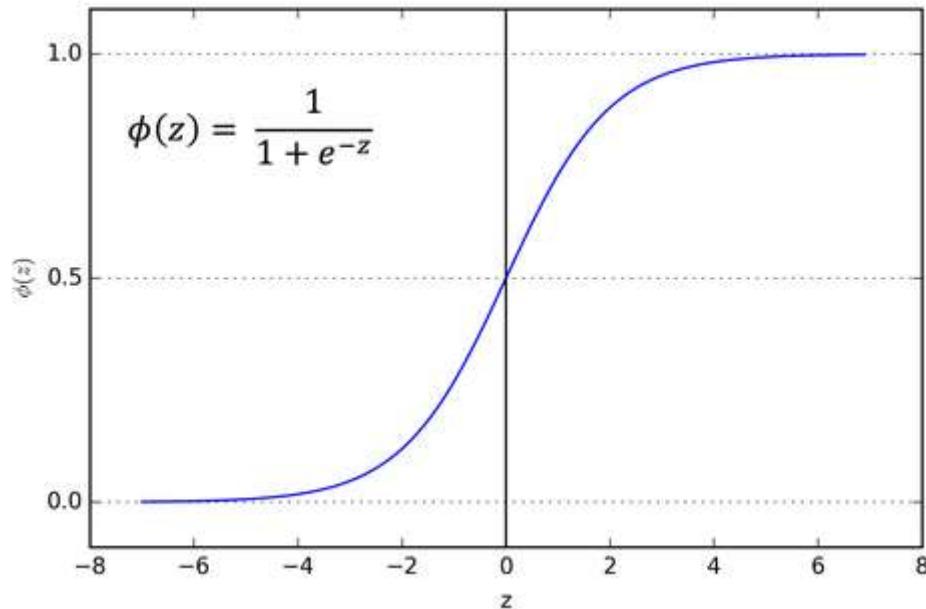


Fig: Sigmoid Function

The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

The function is differentiable. That means, we can find the slope of the sigmoid curve at any two points.

The function is monotonic, but function's derivative is not.

The logistic sigmoid function can cause a neural network to get stuck at the training time.

The softmax function is a more generalized logistic activation function which is used for multiclass classification.

2. Tanh or hyperbolic tangent Activation Function

tanh is also like logistic sigmoid but better. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped).

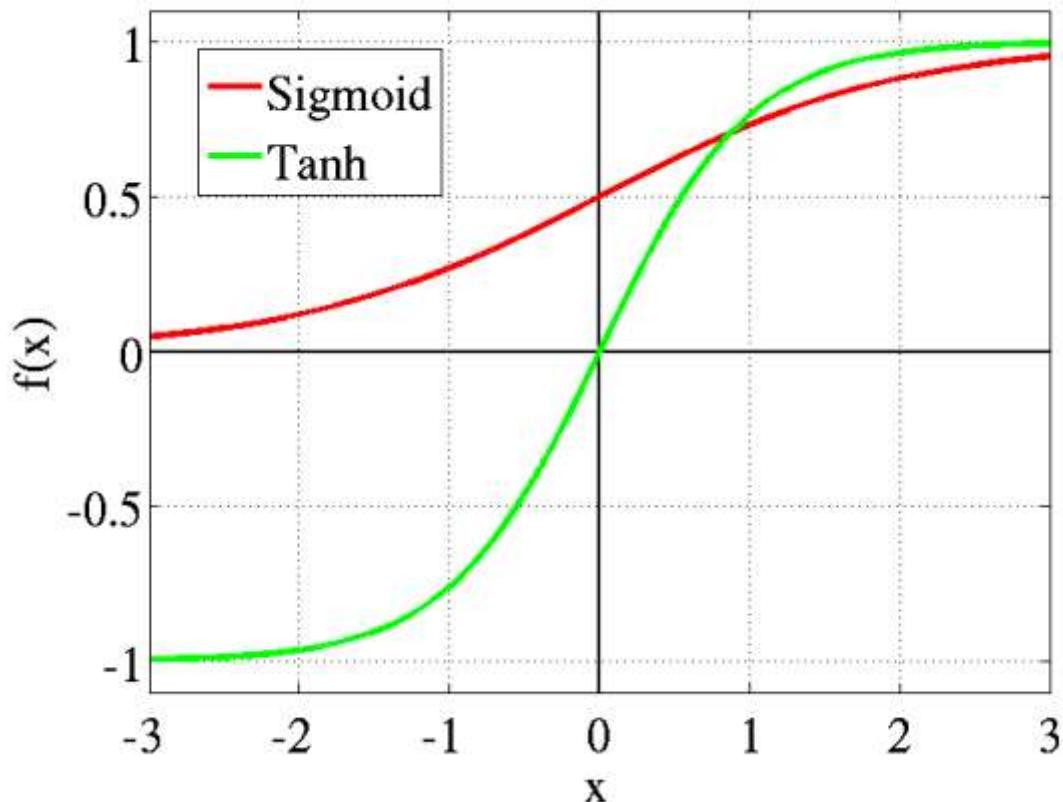


Fig: tanh v/s Logistic Sigmoid

The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph.

The function is differentiable.

The function is monotonic while its derivative is not monotonic.

The tanh function is mainly used classification between two classes.

Both tanh and logistic sigmoid activation functions are used in feed-forward nets.

3. ReLU (Rectified Linear Unit) Activation Function

The ReLU is the most used activation function in the world right now. Since, it is used in almost all the convolutional neural networks or deep learning.

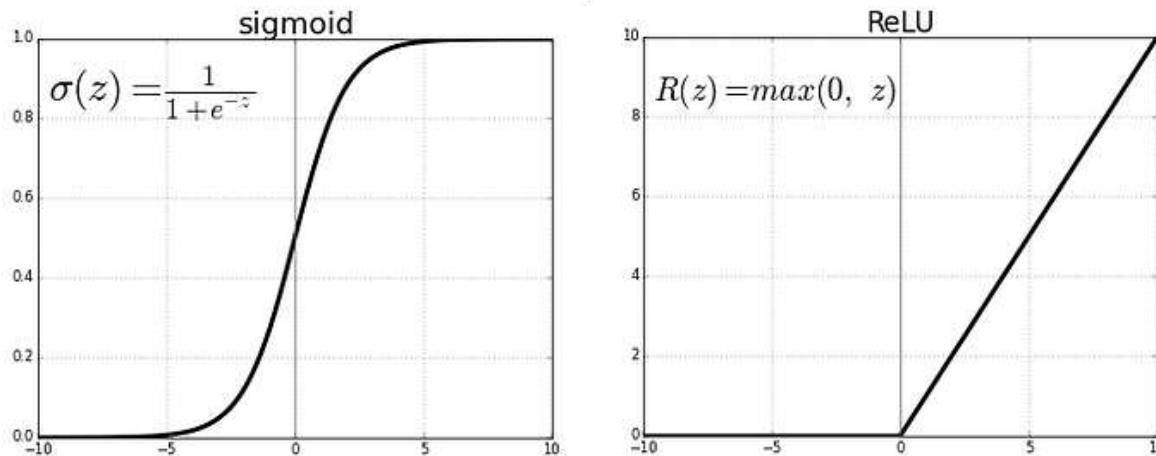


Fig: ReLU v/s Logistic Sigmoid

As you can see, the ReLU is half rectified (from bottom). $f(z)$ is zero when z is less than zero and $f(z)$ is equal to z when z is above or equal to zero.

Range: [0 to infinity)

The function and its derivative both are monotonic.

But the issue is that all the negative values become zero immediately which decreases the ability of the model to fit or train from the data properly. That means any negative input given to the ReLU activation function turns the value into zero immediately in the graph, which in turns affects the resulting graph by not mapping the negative values appropriately.

4. Leaky ReLU

It is an attempt to solve the dying ReLU problem

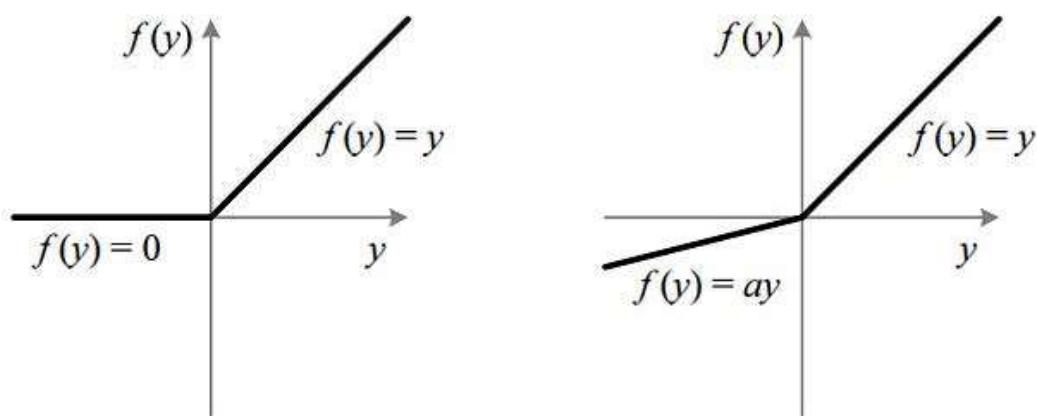


Fig : ReLU v/s Leaky ReLU

Can you see the Leak? 😊

The leak helps to increase the range of the ReLU function. Usually, the value of a is 0.01 or so.

When a is not 0.01 then it is called Randomized ReLU.

Therefore, the range of the Leaky ReLU is (-infinity to infinity).

Both Leaky and Randomized ReLU functions are monotonic in nature. Also, their derivatives also monotonic in nature.

Why is derivative/differentiation used ?

When updating the curve, to know in **which direction** and **how much** to change or update the curve depending upon the slope. That is why we use differentiation in almost every part of Machine Learning and Deep Learning.

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
Arctan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parametric Rectified Linear Unit (PReLU) ^[2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) ^[3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

Fig: Activation Function Cheatsheet

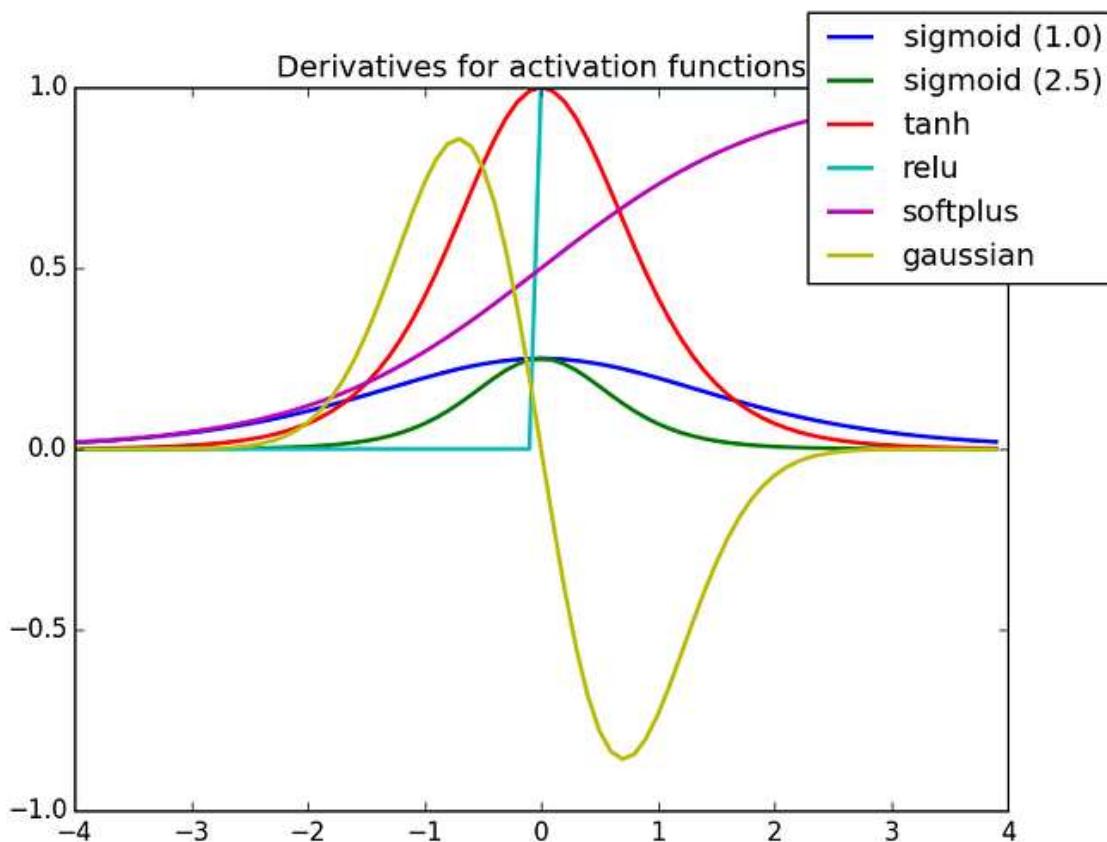


Fig: Derivative of Activation Functions

Sigmoid
 $\sigma(x) = \frac{1}{1+e^{-x}}$



tanh
 $\tanh(x)$



ReLU
 $\max(0, x)$



Leaky ReLU
 $\max(0.1x, x)$



Maxout
 $\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Fig: Different Activation functions
 Reference - <https://hsf-training.github.io/hsf-training-ml-webpage/03-nn/index.html>

Weights

Weights are the real values that are associated with each feature which tells the importance of that feature in predicting the final value.

What do the weights in a Neuron convey to us?

- Importance of the feature

Weights associated with each feature, convey the importance of that feature in predicting the output value. Features with weights that are close to zero said to have lesser importance in the prediction process compared to the features with weights having a larger value.

- Tells the relationship between a particular feature in the dataset and the target value.

Bias

Bias is used for shifting the activation function towards left or right, it can be referred to as a y-intercept in the line equation.

In simple words, neural network bias can be defined as the constant which is added to the product of features and weights. It is used to offset the result. It helps the models to shift the activation function towards the positive or negative side.

Let us understand the importance of bias with the help of an example.

Consider a sigmoid activation function which is represented by the equation below:

$$\text{sigmoid function} = \frac{1}{1 + e^{-x}}$$

On replacing the variable 'x' with the equation of line, we get the following:

$$\text{sigmoid function} = \frac{1}{1 + e^{-(w*x+b)}}$$

In the above equation, 'w' is weights, 'x' is the feature vector, and 'b' is defined as the bias. On substituting the value of 'b' equal to 0, we get the graph of the above equation as shown in the figure below:

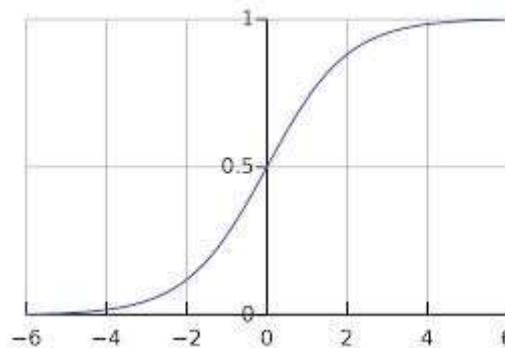


Image source: Wikimedia

If we vary the values of the weight 'w', keeping bias 'b'=0, we will get the following graph:

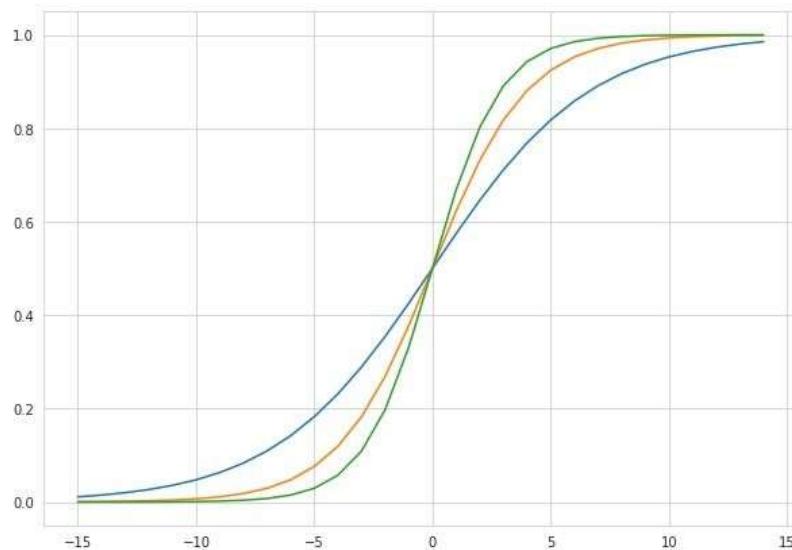


Image source: Medium

While changing the values of 'w', there is no way we can shift the origin of the activation function, i.e., the sigmoid function. On changing the values of 'w', only the steepness of the curve will change. There is only one way to shift the origin and that is to include bias 'b'.

Forward Propagation

Forward propagation is how neural networks make predictions. Input data is “forward propagated” through the network layer by layer to the final layer which outputs a prediction.

Forward propagation (or *forward pass*) refers to the calculation and storage of intermediate variables (including outputs) for a neural network in order from the input layer to the output layer. We now work step-by-step through the mechanics of a neural network with one hidden layer. This may seem tedious but in the eternal words of funk virtuoso James Brown, you must “pay the cost to be the boss”.

For the sake of simplicity, let us assume that the input example is $x \in R^d$ and that our hidden layer does not include a bias term. Here the intermediate variable is:

$$z = W^{(1)}x,$$

where $W^{(1)} \in R^{q \times h}$ is the weight parameter of the hidden layer. After running the intermediate variable $z \in R^h$ through the activation function ϕ we obtain our hidden activation vector of length h ,

$$h = \phi(z)$$

The hidden variable h is also an intermediate variable. Assuming that the parameters of the output layer only possess a weight of $W^{(2)} \in R^{q \times h}$, we can obtain an output layer variable with a vector of length q :

$$ho = W^{(2)}h.$$

Assuming that the loss function is l and the example label is y , we can then calculate the loss term for a single data example,

$$L = l(o, y)$$

According to the definition of L2 regularization, given the hyperparameter λ , the regularization term is

$$s = \frac{\lambda}{2} (\| W^{(1)} \|_F + \| W^{(2)} \|_F)$$

where the Frobenius norm of the matrix is simply the L2 norm applied after flattening the matrix into a vector. Finally, the model’s regularized loss on a given data example is:

$$J = L + s.$$

We refer to J as the objective function in the following discussion.

Backpropagation

In machine learning, backpropagation is a widely used algorithm for training feedforward neural networks. Generalizations of backpropagation exist for other artificial neural networks (ANNs), and for functions generally. These classes of algorithms are all referred to generically as "backpropagation". In fitting a neural network, backpropagation computes the gradient of the loss function with respect to the weights of the network for a single input–output example, and does so efficiently, unlike a naive direct computation of the gradient with respect to each weight individually.

This efficiency makes it feasible to use gradient methods for training multilayer networks, updating weights to minimize loss; gradient descent, or variants such as stochastic gradient descent, are commonly used. The backpropagation algorithm works by computing the gradient of the loss function with respect to each weight by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer to avoid redundant calculations of intermediate terms in the chain rule; this is an example of dynamic programming.

Backpropagation refers to the method of calculating the gradient of neural network parameters. In short, the method traverses the network in reverse order, from the output to the input layer, according to the *chain rule* from calculus.

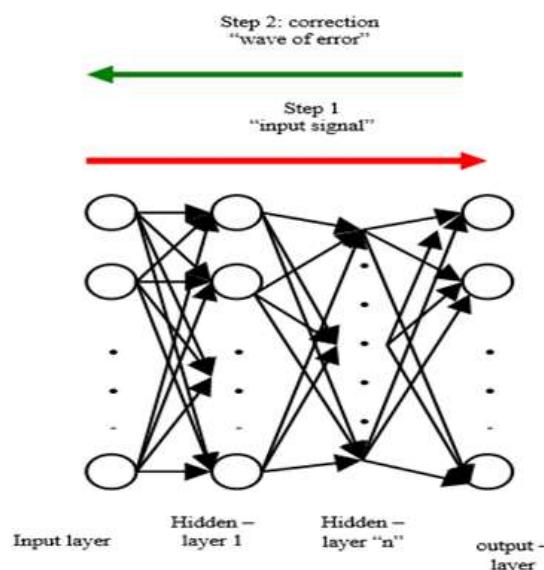


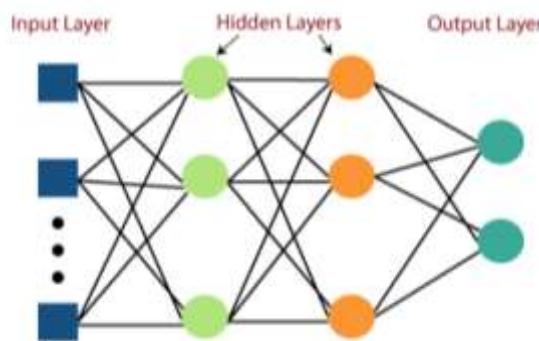
Fig: Forward and backward propagation in Neural Networks

Reference - Jorge Guerra Pires, CC BY-SA 3.0 <<https://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons

1.5 Multilayer Perceptron

Multilayer Perceptron is commonly used in simple regression problems. However, MLPs are not ideal for processing patterns with sequential and multidimensional data.

Multi-Layer perceptron defines the most complex architecture of artificial neural networks. It is substantially formed from multiple layers of the perceptron. TensorFlow is a very popular deep learning framework released by, and this notebook will guide to build a neural network with



Reference -<https://www.javatpoint.com/multi-layer-perceptron-in-tensorflow>

MLP networks are used for supervised learning format. A typical learning algorithm for MLP networks is also called back propagation's algorithm.

A multilayer perceptron (MLP) is a feed forward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input nodes connected as a directed graph between the input and output layers. MLP uses backpropagation for training the network. MLP is a deep learning method.

1.6 Gradient Descent

Introduction

Gradient Descent is known as one of the most commonly used optimization algorithms to train machine learning models by means of minimizing errors between actual and expected results. Further, gradient descent is also used to train Neural Networks. In

mathematical terminology, Optimization algorithm refers to the task of minimizing/maximizing an objective function $f(x)$ parameterized by x .

Gradient descent was initially discovered by "**Augustin-Louis Cauchy**" in the mid-18th century. **Gradient Descent is defined as one of the most commonly used iterative optimization algorithms of machine learning to train the machine learning and deep learning models. It helps in finding the local minimum of a function.**

The best way to define the local minimum or local maximum of a function using gradient descent is as follows:

- If we move towards a negative gradient or away from the gradient of the function at the current point, it will give the local minimum of that function.
- Whenever we move towards a positive gradient or towards the gradient of the function at the current point, we will get the local maximum of that function.

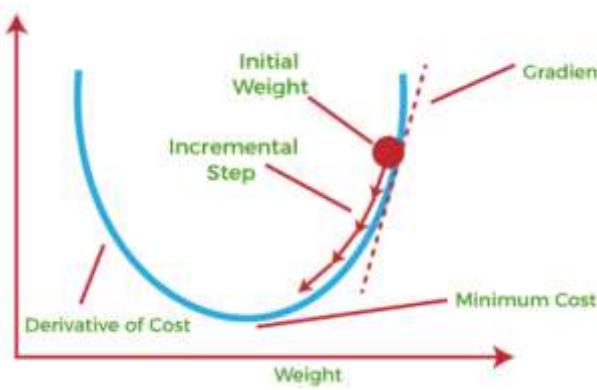


Fig: Gradient Descent

The main objective of using a gradient descent algorithm is to minimize the cost function using iteration. To achieve this goal, it performs two steps iteratively:

- Calculates the first-order derivative of the function to compute the gradient or slope of that function.
- Move away from the direction of the gradient, which means slope increased from the current point by alpha times, where Alpha is defined as Learning Rate. It is a tuning parameter in the optimization process which helps to decide the length of the steps.

Cost Function

The cost function is defined as the measurement of difference or error between actual values and expected values at the current position and present in the form of a single real number. It helps to increase and improve machine learning efficiency by providing feedback to this model so that it can minimize error and find the local or global minimum.

The cost function is calculated after making a hypothesis with initial parameters and modifying these parameters using gradient descent algorithms over known data to reduce the cost function.

How does Gradient Descent work?

Before starting the working principle of gradient descent, we should know some basic concepts to find out the slope of a line from linear regression. The equation for simple linear regression is given as:

$$y = mx + c$$

where 'm' represents the slope of the line, and 'c' represents the intercepts on the y-axis.



Fig: Gradient Descent

The starting point (shown in above fig.) is used to evaluate the performance as it is considered just as an arbitrary point. At this starting point, we will derive the first derivative or slope and then use a tangent line to calculate the steepness of this slope. Further, this slope will inform the updates to the parameters (weights and bias).

Learning Rate

It is defined as the step size taken to reach the minimum or lowest point. This is typically a small value that is evaluated and updated based on the behaviour of the cost function. If the learning rate is high, it results in larger steps but also leads to risks of overshooting the minimum. At the same time, a low learning rate shows the small step sizes, which compromises overall efficiency but gives the advantage of more precision.

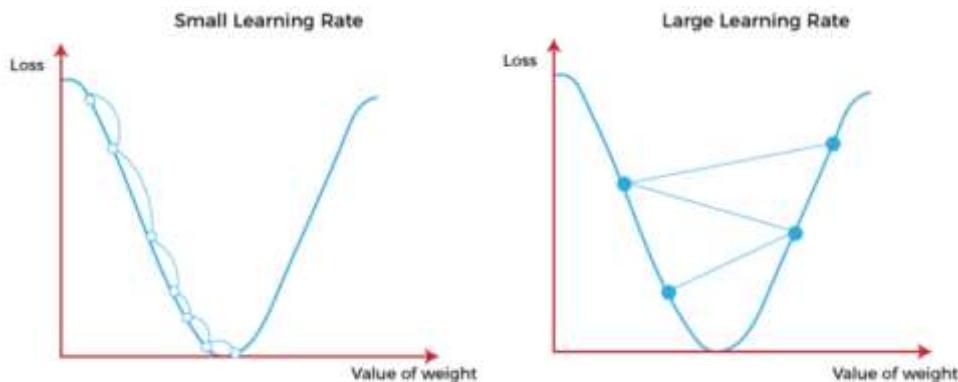


Fig: Learning Rate

Types of Gradient Descent

1. Batch Gradient Descent: Batch gradient descent (BGD) is used to find the error for each point in the training set and update the model after evaluating all training examples. This procedure is known as the training epoch. In simple words, it is a greedy approach where we have to sum over all examples for each update.

Advantages of Batch Gradient Descent:

- It produces less noise in comparison to other gradient descent.
- It produces stable gradient descent convergence.

2. Stochastic Gradient Descent: Stochastic gradient descent (SGD) is a type of gradient descent that runs one training example per iteration. Or in other words, it processes a training epoch for each example within a dataset and updates each training example's parameters one at a time. As it requires only one training example at a time, hence it is easier to store in allocated memory.

Advantages of Stochastic Gradient Descent

- It is easier to allocate in desired memory.
- It is more efficient for large datasets

1. Minibatch Gradient Descent: Mini Batch gradient descent is the combination of both batch gradient descent and stochastic gradient descent. It divides the training datasets into small batch sizes then performs the updates on those batches separately. Splitting training datasets into smaller batches make a balance to maintain the computational efficiency of batch gradient descent and speed of stochastic gradient descent.

Advantages of MiniBatch Gradient Descent

- It is computationally efficient.
- It produces stable gradient descent convergence.

Challenges with Gradient Descent

1. Local Minima & Saddle Point

For convex problems, gradient descent can find the global minimum easily, while for non-convex problems, it is sometimes difficult to find the global minimum, where the machine learning models achieve the best results.

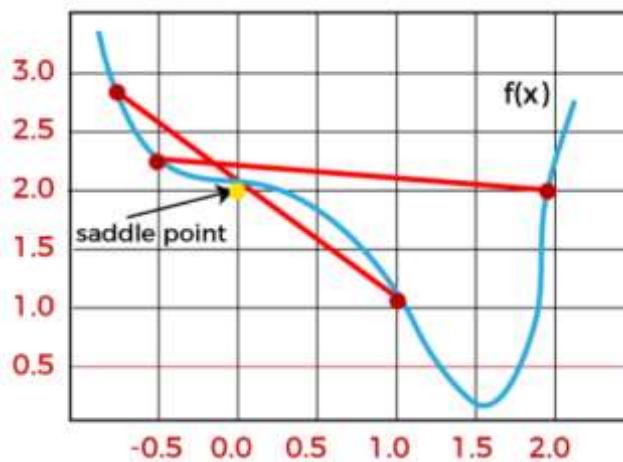


Fig: Saddle point

In contrast, with saddle points, the negative gradient only occurs on one side of the point, which reaches a local maximum on one side and a local minimum on the other side. The name of a saddle point is taken by that of a horse's saddle.

The name of local minima is because the value of the loss function is minimum at that point in a local region. In contrast, the name of the global minima is given so because the value of the loss function is minimum there, globally across the entire domain the loss function.

2. Vanishing and Exploding Gradient

Vanishing Gradients

Vanishing Gradient occurs when the gradient is smaller than expected. During backpropagation, this gradient becomes smaller than causing the decrease in the learning rate of earlier layers than the later layer of the network. Once this happens, the weight parameters update until they become insignificant.

Exploding Gradient

Exploding gradient is just opposite to the vanishing gradient as it occurs when the Gradient is too large and creates a stable model. Further, in this scenario, model weight increases, and they will be represented as NaN. This problem can be solved using the dimensionality reduction technique, which helps to minimize complexity within the model.

1.7 Loss Function

Training Loss

The training loss is a metric used to assess how a deep learning model fits the training data. That is to say, it assesses the error of the model on the training set. Note that, the training set is a portion of a dataset used to initially train the model. Computationally, the training loss is calculated by taking the sum of errors for each example in the training set.

It is also important to note that the training loss is measured after each batch. This is usually visualized by plotting a curve of the training loss.

Validation Loss

On the contrary, validation loss is a metric used to assess the performance of a deep learning model on the validation set. The validation set is a portion of the dataset set aside to validate the performance of the model. The validation loss is similar to the training loss and is calculated from a sum of the errors for each example in the validation set.

Additionally, the validation loss is measured after each epoch. This informs us as to whether the model needs further tuning or adjustments or not. To do this, we usually plot a learning curve for the validation loss

Implications of Training and Validation Loss

In most deep learning projects, the training and validation loss is usually visualized together on a graph. The purpose of this is to diagnose the model's performance and identify which aspects need tuning. To explain this section, we'll use three different scenarios.

Underfitting

Let's consider scenario 1, the image illustrates that the training loss and validation loss are both high:

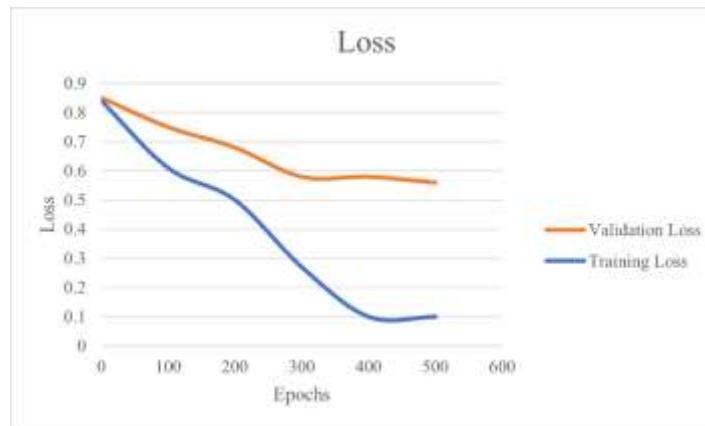


Fig: Underfitting

At times, the validation loss is greater than the training loss. This may indicate that the model is underfitting. Underfitting occurs when the model is unable to accurately model the training data, and hence generates large errors.

Furthermore, the results in scenario 1 indicate that further training is needed to reduce the loss incurred during training. Alternatively, we can also increase the training data either by obtaining more samples or augmenting the data.

Overfitting

In scenario 2, the validation loss is greater than the training loss, as seen in the image:

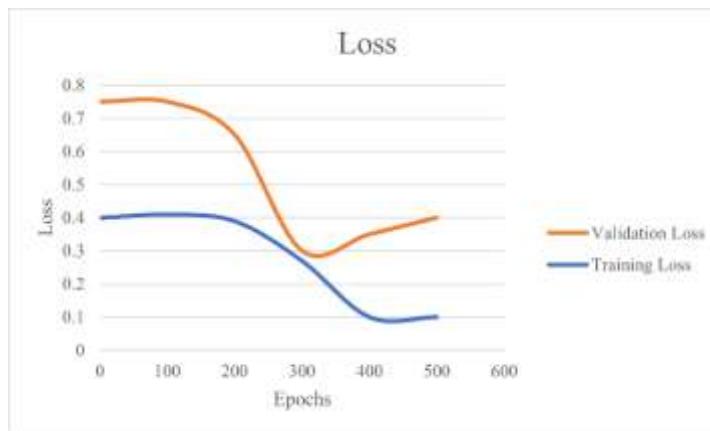


Fig: Overfitting

This usually indicates that the model is overfitting, and cannot generalize on new data. In particular, the model performs well on training data but poorly on the new data in the validation set. At a point, the validation loss decreases but starts to increase again.

A notable reason for this occurrence is that the model may be too complex for the data or that, the model was trained for a long period. In this case, training can be halted when the loss is low and stable, this is usually known as early stopping. Early stopping is one of the many approaches used to prevent overfitting.

Good Fit

In scenario 3, in the image below, the training loss and validation loss both decrease and stabilize at a specific point:

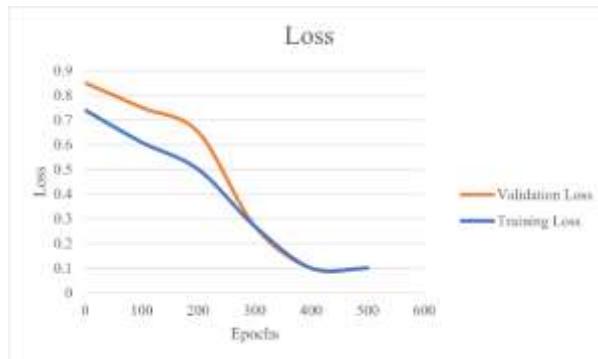


Fig: Good Fit

This indicates an optimal fit, i.e a model that does not overfit or underfit.

TensorFlow 2.0 and Keras API

TensorFlow is one of the most popular and widely used Deep Learning libraries in the companies these days. It helps you work with complex data and build neural network models to solve business problems.

TensorFlow 2.0

TensorFlow 2.0 is a library that provides a comprehensive ecosystem of tools for developers, researchers, and organizations who want to build scalable Machine Learning and Deep Learning applications.

TensorFlow is a popular open-source library released in 2015 by the Google Brain team for building machine learning and deep learning models. It is based on Python programming language and performs numerical computations using data flow graphs to build models.

Features of TensorFlow 2.0

- TensorFlow 2.0 supports easy model building with Keras and eager execution.
- It has robust model deployment in production on any platform.
- You can perform robust experimentation for research.
- TensorFlow 2.0 simplifies the API by cleaning up deprecated APIs and reducing duplication.
- TensorFlow 2.0 works efficiently with multi-dimensional arrays.
- TensorFlow 2.0 supports fast debugging and model building.

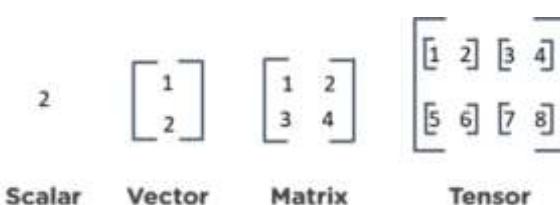
TensorFlow Applications

Below are a few examples where TensorFlow is being widely used:

- Face detection in electronic devices.
- Machine language translation through apps such as Google Translate.
- Fraud detection in the banking and financial sectors.
- Object detections on videos.

Tensor

TensorFlow is derived from its core component known as a tensor. A tensor is actually a vector or a matrix of n-dimensions that represents all types of data.



In TensorFlow, we define tensors by a unit of dimensionality known as a rank.

	Example	Entity	Rank
s = 300		Scalar	0
v = [1, 2, 3]		Vector	1
m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]		Matrix	2
t = [[[2], [4], [6]], [[8], [9], [10]], [[11], [12], [13]]]		Tensor	3

TensorFlow performs computations with the help of dataflow graphs. It has nodes that represent the operations in your model.

Let's compute the function depicted below and see how TensorFlow works:

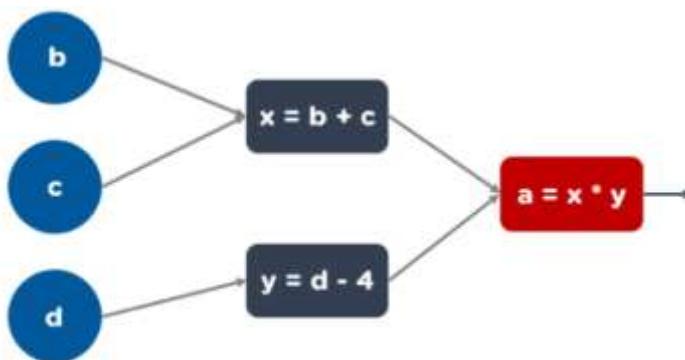
$$a(b, c, d) = (b + c) * (d - 4)$$

$$x = b + c$$

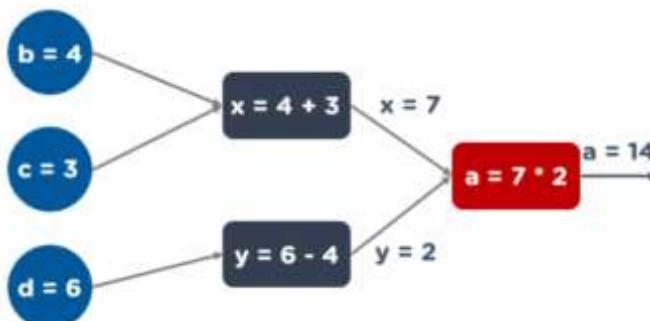
$$y = d - 4$$

$$a = x * y$$

The graph nodes are the inputs and perform mathematical computations, while the connections carry the weights. In this case, it's the result of an expression.



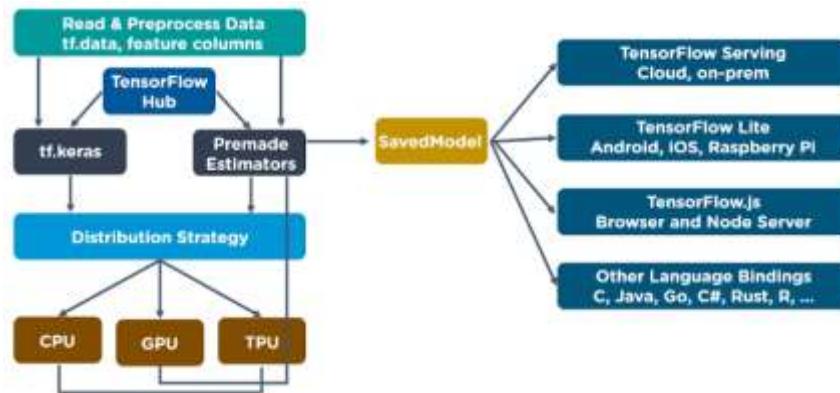
Displayed below is the output:



TensorFlow 2.0 Architecture

Over the last few years, the developer community has added many components to TensorFlow. These components will be packaged together into a comprehensive platform that supports machine learning workflows from training through deployment.

Shown below is the new architecture of TensorFlow 2.0:



Keras API

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

KERAS IS Simple -- but not simplistic. Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter.

Flexible -- Keras adopts the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what you've already learned.

Powerful -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.

Keras & TensorFlow 2

TensorFlow 2 is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

- Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
- Computing the gradient of arbitrary differentiable expressions.
- Scaling computation to many devices, such as clusters of hundreds of GPUs.
- Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep

learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.N5T

Unit 2: Computer Vision Basics

Learning Outcomes:

- Understand basic key concepts of computer vision
- Understand concept of different layers in Convolution neural network
- Implement practical of computer vision using openCV

2.1 What is Computer Vision (CV)?

Computer vision (CV) is an artificial intelligence (AI) subcategory that focuses on developing and deploying digital systems that process, analyze, and interpret visual input.

The objective of computer vision is to allow computers to recognize an item or person in a digital image and take appropriate action.

Convolutional neural networks (CNNs) are used in computer vision to analyze visual input at the pixel level.

Difference between Image Processing and Computer Vision:

Image Processing	Computer Vision
Image processing is mainly focused on processing the raw input images to enhance them or preparing them to do other tasks	Computer vision is focused on extracting information from the input images or videos to have a proper understanding of them to predict the visual input like human brain.
Image processing uses methods like Anisotropic diffusion, Hidden Markov models, Independent component analysis, Different Filtering etc.	Image processing is one of the methods that is used for computer vision along with other Machine learning techniques, CNN etc.
Image Processing is a subset of Computer Vision	Computer Vision is a superset of Image Processing.
Examples of some Image Processing applications are- Rescaling image (Digital Zoom), Correcting illumination, Changing tones etc.	Examples of some Computer Vision applications are- Object detection, Face detection, Hand writing recognition etc.

2.2 Image Fundamentals:

What's an image?

An image refers to a 2D light intensity function $f(x,y)$, where (x,y) denote spatial coordinates and the value of f at any point (x,y) is proportional to the brightness or gray levels of the image at that point.

A digital image is an image $f(x,y)$ that has been discretized both in spatial coordinates and brightness.

The elements of such a digital array are called image elements or pixels.

Types of Images



{a}

Color Image



{b}

Grayscale Image



{c}

Binary Image

Pixels

A pixel is the smallest unit of a digital image or graphic that can be displayed and represented on a digital display device. A pixel is the basic logical unit in digital graphics. Pixels are combined to form a complete image, video, text, or any visible thing on a computer display.

Gray-scale images

Grayscale images are monochrome images, Means they have only one color. Grayscale images do not contain any information about color. Each pixel determines available different grey levels.

A normal grayscale image contains 8 bits/pixel data, which has 256 different grey levels. In medical images and astronomy, 12 or 16 bits/pixel images are used.

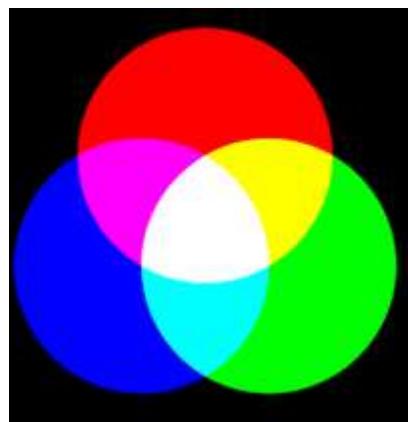


Reference: <https://www.javatpoint.com/dip-types-of-images>

Colour images

Colour images are three band monochrome images in which, each band contains a different color and the actual information is stored in the digital image. The color images contain gray level information in each spectral band.

The images are represented as red, green and blue (RGB images). And each color image has 24 bits/pixel means 8 bits for each of the three color band(RGB).



Reference: <https://www.javatpoint.com/dip-types-of-images>

2.3 Computer Vision Using OpenCV

OpenCV is a very famous library for computer vision and image processing tasks. It one of the most used pythons open-source library for computer vision and image data.

It is used in various tasks such as image denoising, image thresholding, edge detection, corner detection, contours, image pyramids, image segmentation, face detection and many more.

Let's understand computer vision with Practical IMPORT LIBRARIES

```
import os
import numpy as np
import cv2
import matplotlib.pyplot as plt
```

RGB IMAGE AND RESIZING

An RGB image where RGB indicates Red, Green, and Blue respectively can be considered as three images stacked on top of each other. It also has a nickname called ‘True Color Image’ as it represents a real-life image as close as possible and is based on human perception of colors.

```
height = 224
width = 224
font_size = 20
plt.figure(figsize=(15, 8))
for i, path in enumerate(paths):
    name = os.path.split(path)[-1]
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    resized_img = cv2.resize(img, (height, width))
    plt.subplot(1, 2, i+1).set_title(name[ : -4], fontsize = font_size); plt.axis('off')
    plt.imshow(cv2.cvtColor(resized_img, cv2.COLOR_BGR2RGB))
plt.show()
```

GRAYSCALE IMAGE

Grayscale images are images that are shades of grey. It represents the degree of luminosity and carries the intensity information of pixels in the image. Black is the weakest intensity and white is the strongest intensity.

```
plt.figure(figsize=(15, 8))
for i, path in enumerate(paths):
    name = os.path.split(path)[-1]
```

```
img = cv2.imread(path, 0)
resized_img = cv2.resize(img, (height, width))

plt.subplot(1, 2, i + 1).set_title(f'Grayscale {name[:-4]} Image', fontsize = font_size);
plt.axis('off')

plt.imshow(resized_img, cmap='gray')
plt.show()
```

IMAGE DENOISING

Image denoising removes noise from the image. It is also known as ‘Image Smoothing’. The image is convolved with a low pass filter kernel which gets rid of high-frequency content like edges of an image

```
for i, path in enumerate(paths):
    name = os.path.split(path)[-1]
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    resized_img = cv2.resize(img, (height, width))
    denoised_img = cv2.medianBlur(resized_img, 5)

    plt.figure(figsize=(15, 8))

    plt.subplot(1, 2, 1).set_title(f'Original {name[:-4]} Image', fontsize = font_size);
    plt.axis('off')

    plt.imshow(cv2.cvtColor(resized_img, cv2.COLOR_BGR2RGB))

    plt.subplot(1, 2, 2).set_title(f'After Median Filtering of {name[:-4]} Image', fontsize = font_size); plt.axis('off')

    plt.imshow(cv2.cvtColor(denoised_img, cv2.COLOR_BGR2RGB))

    plt.show()
```

IMAGE THRESHOLDING

Image Thresholding is self-explanatory. If the pixel value in an image is above a certain threshold, a particular value is assigned and if it is below the threshold, another particular value is assigned.

```
for i, path in enumerate(paths):
    name = os.path.split(path)[-1]
    img = cv2.imread(path, 0)
```

```

resized_img = cv2.resize(img, (height, width))

denoised_img = cv2.medianBlur(resized_img, 5)

th = cv2.adaptiveThreshold(denoised_img, maxValue = 255, adaptiveMethod =
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, thresholdType = cv2.THRESH_BINARY,
blockSize = 11, C = 2)

plt.figure(figsize=(15, 8))

plt.subplot(1, 2, 1).set_title(f'Grayscale {name[ : -4]} Image', fontsize = font_size);
plt.axis('off')

plt.imshow(resized_img, cmap = 'gray')

plt.subplot(1, 2, 2).set_title(f'After Adapative Thresholding of {name[ : -4]} Image',
fontsize = font_size); plt.axis('off')

plt.imshow(cv2.cvtColor(th, cv2.COLOR_BGR2RGB))

plt.show()

```

IMAGE GRADIENTS

Gradients are the slope of the tangent of the graph of the function. Image gradients find the edges of a grayscale image in the x and y-direction. This can be done by calculating derivates in both directions using Sobel x and Sobel y operations.

```

for i, path in enumerate(paths):

    name = os.path.split(path)[-1]

    img = cv2.imread(path, 0)

    resized_img = cv2.resize(img, (height, width))

    laplacian = cv2.Laplacian(resized_img, cv2.CV_64F)

    plt.figure(figsize=(15, 8))

    plt.subplot(1, 2, 1).set_title(f'Grayscale {name[ : -4]} Image', fontsize = font_size);
    plt.axis('off')

    plt.imshow(resized_img, cmap = 'gray')

    plt.subplot(1, 2, 2).set_title(f'After finding Laplacian Derivatives of {name[ : -4]} Image',
    fontsize = font_size); plt.axis('off')

    plt.imshow(cv2.cvtColor(laplacian.astype('float32'), cv2.COLOR_BGR2RGB))

    plt.show()

```

EDGE DETECTION

Edge Detection is performed using Canny Edge Detection which is a multi-stage algorithm. The stages to achieve edge detection are as follows. Noise Reduction – Smoothen image using Gaussian filter

Find Intensity Gradient – Using the Sobel kernel, find the first derivative in the horizontal (G_x) and vertical (G_y) directions.

```
for i, path in enumerate(paths):
```

```
    name = os.path.split(path)[-1]
```

```
    img = cv2.imread(path, 0)
```

```
    resized_img = cv2.resize(img, (height, width))
```

```
    edges = cv2.Canny(resized_img, threshold1 = 100, threshold2 = 200)
```

```
    plt.figure(figsize=(15, 8))
```

```
    plt.subplot(1, 2, 1).set_title(f'Grayscale {name[ : -4]} Image', fontsize = font_size);  
    plt.axis('off')
```

```
    plt.imshow(resized_img, cmap = 'gray')
```

```
    plt.subplot(1, 2, 2).set_title(f'After Canny Edge Detection of {name[ : -4]} Image',  
        fontsize = font_size); plt.axis('off')
```

```
    plt.imshow(cv2.cvtColor(edges, cv2.COLOR_BGR2RGB))
```

```
    plt.show()
```

FOURIER TRANSFORM ON IMAGE

Fourier Transform analyzes the frequency characteristics of an image. Discrete Fourier Transform is used to find the frequency domain.

```
for i, path in enumerate(paths):
```

```
    name = os.path.split(path)[-1]
```

```
    img = cv2.imread(path, 0)
```

```
    resized_img = cv2.resize(img, (height, width))
```

```
    freq = np.fft.fft2(resized_img)
```

```
    freq_shift = np.fft.fftshift(freq)
```

```
    magnitude_spectrum = 20 * np.log(np.abs(freq_shift))
```

```
    plt.figure(figsize=(15, 8))
```

```
    plt.subplot(1, 2, 1).set_title(f'Grayscale {name[ : -4]} Image', fontsize = font_size);  
    plt.axis('off')
```

```

plt.imshow(cv2.cvtColor(resized_img, cv2.COLOR_BGR2RGB))

plt.subplot(1, 2, 2).set_title(f'Magnitude Spectrum of {name[ : -4]} Image', fontsize = font_size); plt.axis('off')

plt.imshow(magnitude_spectrum, cmap = 'gray')

plt.show()

```

MORPHOLOGICAL TRANSFORMATION OF IMAGE

Morphological Transformation is usually applied on binary images where it takes an image and a kernel which is a structuring element as inputs. Binary images may contain imperfections like texture and noise.

These transformations help in correcting these imperfections by accounting for the form of the image

```

kernel = np.ones((5,5), np.uint8)

plt.figure(figsize=(15, 8))

img = cv2.imread('../input/cv-images/morph-min.jpg', cv2.IMREAD_COLOR)

resized_img = cv2.resize(img, (height, width))

morph_open = cv2.morphologyEx(resized_img, cv2.MORPH_OPEN, kernel)

morph_close = cv2.morphologyEx(morph_open, cv2.MORPH_CLOSE, kernel)

plt.subplot(1,2,1).set_title('Original Digit - 7 Image', fontsize = font_size); plt.axis('off')

plt.imshow(cv2.cvtColor(resized_img, cv2.COLOR_BGR2RGB))

plt.subplot(1,2,2).set_title('After Morphological Opening and Closing of Digit - 7 Image', fontsize = font_size); plt.axis('off')

plt.imshow(cv2.cvtColor(morph_close, cv2.COLOR_BGR2RGB))

plt.show()

```

GEOMETRIC TRANSFORMATION OF IMAGE

Geometric Transformation of images is achieved by two transformation functions namely cv2.warpAffine and cv2.warpPerspective that receive a 2×3 and 3×3 transformation matrix respectively.

```

pts1 = np.float32([[1550, 1170],[2850, 1370],[50, 2600],[1850, 3450]])

pts2 = np.float32([[0,0],[4160,0],[0,3120],[4160,3120]])

img = cv2.imread('../input/cv-images/book-min.jpg', cv2.IMREAD_COLOR)

```

```

transformation_matrix = cv2.getPerspectiveTransform(pts1, pts2)

final_img = cv2.warpPerspective(img, M = transformation_matrix, dsize = (4160,
3120))

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

img = cv2.resize(img, (256, 256))

final_img = cv2.cvtColor(final_img, cv2.COLOR_BGR2RGB)

final_img = cv2.resize(final_img, (256, 256))

plt.figure(figsize=(15, 8))

plt.subplot(1,2,1).set_title('Original Book Image', fontsize = font_size); plt.axis('off')

plt.imshow(img)

plt.subplot(1,2,2).set_title('After Perspective Transformation of Book Image', fontsize
= font_size); plt.axis('off')

plt.imshow(final_img)

plt.show()

```

2.4 Practical: Canny Edge Detection

Canny edge detection is a image processing method used to detect edges in an image while suppressing noise. The main steps are as follows:

Algorithm Steps

- Grayscale conversion
- Gaussian blur
- Determine the intensity gradients
- Non maximum suppression
- Double thresholding
- Edge tracking by hysteresis
- Cleaning up

Code

```

import org.opencv.core.Core;

import org.opencv.core.Mat;

import org.opencv.imgcodecs.Imgcodecs;

```

```
import org.opencv.imgproc.Imgproc;

public class CannyEdgeDetection {
    public static void main(String args[]) throws Exception {
        // Loading the OpenCV core library
        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);

        // Reading the Image from the file and storing it in to a Matrix object
        String file = "E:/OpenCV/chap17/canny_input.jpg";

        // Reading the image
        Mat src = Imgcodecs.imread(file);

        // Creating an empty matrix to store the result
        Mat gray = new Mat();

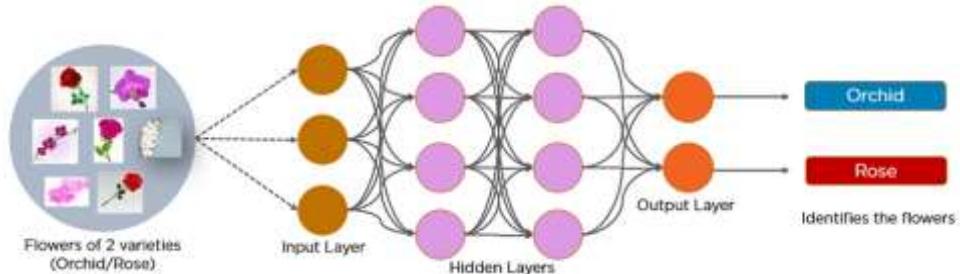
        // Converting the image from color to Gray
        Imgproc.cvtColor(src, gray, Imgproc.COLOR_BGR2GRAY);
        Mat edges = new Mat();

        // Detecting the edges
        Imgproc.Canny(gray, edges, 60, 60*3);

        // Writing the image
        Imgcodecs.imwrite("E:/OpenCV/chap17/canny_output.jpg", edges);
        System.out.println("Image Loaded");
    }
}
```

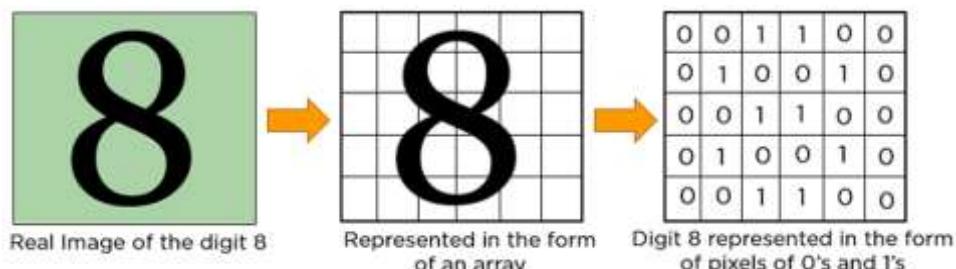
2.5 Convolutional Neural Network

A convolutional neural network is a feed-forward neural network that is generally used to analyze visual images by processing data with grid-like topology. It's also known as a ConvNet. A convolutional neural network is used to detect and classify objects in an image.



Reference: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network>

In CNN, every image is represented in the form of an array of pixel values



2.6 Explanation of Convolutional Neural Network in detail

A convolution neural network has multiple hidden layers that help in extracting information from an image. The four important layers in CNN are:

1. Convolution layer
2. ReLU layer
3. Pooling layer
4. Fully connected layer

Convolution Layer

This is the first step in the process of extracting valuable features from an image. A convolution layer has several filters that perform the convolution operation. Every image is considered as a matrix of pixel values.

Consider the following 5x5 image whose pixel values are either 0 or 1. There's also a filter matrix with a dimension of 3x3. Slide the filter matrix over the image and compute the dot product to get the convolved feature matrix

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature. After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

ReLU layer

ReLU stands for the rectified linear unit. Once the feature maps are extracted, the next step is to move them to a ReLU layer.

ReLU performs an element-wise operation and sets all the negative pixels to 0. It introduces non-linearity to the network, and the generated output is a rectified feature map. Below is the graph of a ReLU function:

Pooling Layer

Pooling layers, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

- Max pooling: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.
- Average pooling: As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

While a lot of information is lost in the pooling layer, it also has a number of benefits to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

Fully-Connected Layer

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1.

Code

```
#importing the required libraries
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPool2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
#loading data
(X_train,y_train) , (X_test,y_test)=mnist.load_data()
#reshaping data
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], X_train.shape[2], 1))
X_test = X_test.reshape((X_test.shape[0],X_test.shape[1],X_test.shape[2],1))
#checking the shape after reshaping
print(X_train.shape)
print(X_test.shape)
#normalizing the pixel values
X_train=X_train/255
X_test=X_test/255
#defineing model
model=Sequential()
#adding convolution layer
model.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
#adding pooling layer
model.add(MaxPool2D(2,2))
#adding fully connected layer
model.add(Flatten())
model.add(Dense(100,activation='relu'))
#adding output layer
model.add(Dense(10,activation='softmax'))
#compiling the model
```

```

model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

#fitting the model

model.fit(X_train,y_train,epochs=10)

```

2.7 Transfer Learning

Transfer learning involves the approach in which knowledge learned in one or more source tasks is transferred and used to improve the learning of a related target task. While most machine learning algorithms are designed to address single tasks, the development of algorithms that facilitate transfer learning is a topic of ongoing interest in the machine-learning community.

Why transfer learning?

Many deep neural networks trained on natural images exhibit a curious phenomenon in common: on the first layer they learn features similar to Gabor filters and color blobs. Such first-layer features appear not to specific to a particular dataset or task but are general in that they are applicable to many datasets and tasks. As finding these standard features on the first layer seems to occur regardless of the exact cost function and natural image dataset, we call these first-layer features general. For example, in a network with an N-dimensional softmax output layer that has been successfully trained towards a supervised classification objective, each output unit will be specific to a particular class. We thus call the last-layer features specific.

In transfer learning we first train a base network on a base dataset and task, and then we repurpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. This process will tend to work if the features are general, that is, suitable to both base and target tasks, instead of being specific to the base task.

In practice, very few people train an entire Convolutional Network from scratch because it is relatively rare to have a dataset of sufficient size. Instead, it is common to pre-train a ConvNet on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then use the ConvNet either as an initialization or a fixed feature extractor for the task of interest.

Transfer learning scenarios

Depending on both the size of the new dataset and the similarity of the new dataset to the original dataset, the approach for using transfer learning will be different. Keeping in mind that ConvNet features are more generic in the early layers and more original-dataset specific in the later layers, here are some common rules of thumb for navigating the four major scenarios:

- The target dataset is small and similar to the base training dataset.
Since the target dataset is small, it is not a good idea to fine-tune the ConvNet due to the risk of overfitting. Since the target data is similar to the base data, we expect higher-level features in the ConvNet to be relevant to this dataset as well. Hence, we:
 - Remove the fully connected layers near the end of the pretrained *base* ConvNet
 - Add a new fully connected layer that matches the number of classes in the *target* dataset
 - Randomize the weights of the new fully connected layer and freeze all the weights from the pre-trained network
 - Train the network to update the weights of the new fully connected layers
- The target dataset is large and similar to the base training dataset.
Since the target dataset is large, we have more confidence that we won't overfit if we try to fine-tune through the full network. Therefore, we:
 - Remove the last fully connected layer and replace with the layer matching the number of classes in the target dataset
 - Randomly initialize the weights in the new fully connected layer
 - Initialize the rest of the weights using the pre-trained weights, i.e., unfreeze the layers of the pre-trained network
 - Retrain the entire neural network
- The target dataset is small and different from the base training dataset.
Since the data is small, overfitting is a concern. Hence, we train only the linear layers. But as the target dataset is very different from the base dataset, the higher level features in the ConvNet would not be of any relevance to the target dataset. So, the new network will only use the lower level features of the base ConvNet. To implement this scheme, we:
 - Remove most of the pre-trained layers near the beginning of the ConvNet
 - Add to the remaining pre-trained layers new fully connected layers that match the number of classes in the new dataset
 - Randomize the weights of the new fully connected layers and freeze all the weights from the pre-trained network
 - Train the network to update the weights of the new fully connected layers
- The target dataset is large and different from the base training dataset.
As the target dataset is large and different from the base dataset, we can train the ConvNet from scratch. However, in practice, it is beneficial to initialize the weights from the pre-trained network and fine-tune them as it

might make the training faster. In this condition, the implementation is the same as in case 3.

2.8 Transfer Learning Hands on

We will be using the Cifar-10 dataset and the keras framework to implement our model. In this post, we will first build a model from scratch and then try to improve it by implementing transfer learning. Before we start to code, let's discuss the Cifar-10 dataset in brief. Cifar-10 dataset consists of 60,000 32*32 color images in 10 classes, with 6000 images per class. There are 50,000 training images and 10,000 testing images. Let's begin by importing the dataset. Since this dataset is present in the keras database, we will import it from keras directly.

```
import numpy as np
from keras.datasets import cifar10

#Load the dataset:
(X_train, y_train), (X_test, y_test) = cifar10.load_data()

print("There are {} train images and {} test images.".format(X_train.shape[0],
X_test.shape[0]))
print('There are {} unique classes to predict.'.format(np.unique(y_train).shape[0]))

#One-hot encoding the labels
num_classes = 10
from keras.utils import np_utils
y_train = np_utils.to_categorical(y_train, num_classes)
y_test = np_utils.to_categorical(y_test, num_classes)

fig = plt.figure(figsize=(10, 10))

for i in range(1, 9):
    img = X_train[i-1]
    fig.add_subplot(2, 4, i)
```

```
plt.imshow(img)
```

```
print(\u2018Shape of each image in the training data: \u2019, X_train.shape[1:])
```

```
#Importing the necessary libraries
```

```
from keras.models import Sequential  
from keras.layers import Dense, Conv2D, MaxPooling2D  
from keras.layers import Dropout, Flatten, GlobalAveragePooling2D
```

```
#Building up a Sequential model
```

```
model = Sequential()  
model.add(Conv2D(32, (3, 3), activation='relu', input_shape = X_train.shape[1:]))  
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Conv2D(32, (3, 3), activation='relu'))  
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Conv2D(64, (3, 3), activation='relu'))  
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(GlobalAveragePooling2D())  
model.add(Dense(10, activation='softmax'))  
model.summary()
```

```
model.compile(loss='binary_crossentropy', optimizer='adam',  
metrics=['accuracy'])
```

```
X_train_scratch = X_train/255.
```

```
X_test_scratch = X_test/255.
```

```
#Creating a checkpointer
checkpointer = ModelCheckpoint(filepath='scratchmodel.best.hdf5',
                               verbose=1,save_best_only=True)

#Fitting the model on the train data and labels.
model.fit(X_train, y_train, batch_size=32, epochs=10,
           verbose=1, callbacks=[checkpointer], validation_split=0.2, shuffle=True)

#Evaluate the model on the test data
score = model.evaluate(X_test, y_test)

#Accuracy on test data
print('Accuracy on the Test Images: ', score[1])

#So, our CNN model produces an accuracy of 82% on the test dataset. implement
transfer learning and check if we can improve the model. We will be using the
Resnet50 model

#Importing the ResNet50 model
from keras.applications.resnet50 import ResNet50, preprocess_input

#Loading the ResNet50 model with pre-trained ImageNet weights
model = ResNet50(weights='imagenet', include_top=False, input_shape=(200, 200,
3))

#Reshaping the training data
X_train_new = np.array([imresize(X_train[i], (200, 200, 3)) for i in range(0,
len(X_train))]).astype('float32')

#Preprocessing the data, so that it can be fed to the pre-trained ResNet50 model.
resnet_train_input = preprocess_input(X_train_new)
```

```
#Creating bottleneck features for the training data
train_features = model.predict(resnet_train_input)

#Saving the bottleneck features
np.savez('resnet_features_train', features=train_features)

#Reshaping the testing data
X_test_new = np.array([imresize(X_test[i], (200, 200, 3)) for i in range(0,
len(X_test))]).astype('float32')

#Preprocessing the data, so that it can be fed to the pre-trained ResNet50 model.
resnet_test_input = preprocess_input(X_test_new)

#Creating bottleneck features for the testing data
test_features = model.predict(resnet_test_input)

#Saving the bottleneck features
np.savez('resnet_features_test', features=test_features)

model = Sequential()
model.add(GlobalAveragePooling2D(input_shape=train_features.shape[1:]))
model.add(Dropout(0.3))
model.add(Dense(10, activation='softmax'))
model.summary()
model.compile(loss='categorical_crossentropy', optimizer='adam',
              metrics=['accuracy'])

model.fit(train_features, y_train, batch_size=32, epochs=10,
          validation_split=0.2, callbacks=[checkpointer], verbose=1, shuffle=True)

#Evaluate the model on the test data
score = model.evaluate(test_features, y_test)

#Accuracy on test data
print('Accuracy on the Test Images: ', score[1])
```

Unit 3: Computer Vision Hands-On

Learning Outcomes:

- Understand concept of different layers in Convolution neural network
- Implement practical of computer vison using openCV

3.1 What is face detection?

Object detection is one of the computer technologies that is connected to image processing and computer vision. It is concerned with detecting instances of an object such as human faces, buildings, trees, cars, etc. The primary aim of face detection algorithms is to determine whether there is any face in an image or not.

In recent years, we have seen significant advancement of technologies that can detect and recognise faces. Our mobile cameras are often equipped with such technology where we can see a box around the faces. Although there are quite advanced face detection algorithms, especially with the introduction of deep learning, the introduction of viola jones algorithm in 2001 was a breakthrough in this field. Now let us explore the viola jones algorithm in detail.

3.2 What is Viola Jones algorithm?

Viola Jones algorithm is named after two computer vision researchers who proposed the method in 2001, Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features”. Despite being an outdated framework, Viola-Jones is quite powerful, and its application has proven to be exceptionally notable in real-time face detection. This algorithm is painfully slow to train but can detect faces in real-time with impressive speed.

Given an image(this algorithm works on grayscale image), the algorithm looks at many smaller subregions and tries to find a face by looking for specific features in each subregion. It needs to check many different positions and scales because an image can contain many faces of various sizes. Viola and Jones used Haar-like features to detect faces in this algorithm.

The Viola Jones algorithm has four main steps, which we shall discuss in the sections to follow:

- Selecting Haar-like features
- Creating an integral image
- Running AdaBoost training

- Creating classifier cascades

What are Haar-Like Features?

In the 19th century a Hungarian mathematician, Alfred Haar gave the concepts of Haar wavelets, which are a sequence of rescaled “square-shaped” functions which together form a wavelet family or basis. Viola and Jones adapted the idea of using Haar wavelets and developed the so-called Haar-like features.

Haar-like features are digital image features used in object recognition. All human faces share some universal properties of the human face like the eyes region is darker than its neighbour pixels, and the nose region is brighter than the eye region.

A simple way to find out which region is lighter or darker is to sum up the pixel values of both regions and compare them. The sum of pixel values in the darker region will be smaller than the sum of pixels in the lighter region. If one side is lighter than the other, it may be an edge of an eyebrow or sometimes the middle portion may be shinier than the surrounding boxes, which can be interpreted as a nose. This can be accomplished using Haar-like features and with the help of them, we can interpret the different parts of a face.

There are 3 types of Haar-like features that Viola and Jones identified in their research:

- Edge features
- Line-features
- Four-sided features

Edge features and Line features are useful for detecting edges and lines respectively. The four-sided features are used for finding diagonal features.

The value of the feature is calculated as a single number: the sum of pixel values in the black area minus the sum of pixel values in the white area. The value is zero for a plain surface in which all the pixels have the same value, and thus, provide no useful information.

Since our faces are of complex shapes with darker and brighter spots, a Haar-like feature gives you a large number when the areas in the black and white rectangles are very different. Using this value, we get a piece of valid information out of the image.

To be useful, a Haar-like feature needs to give you a large number, meaning that the areas in the black and white rectangles are very different. There are known features that perform very well to detect human faces:

For example, when we apply this specific haar-like feature to the bridge of the nose, we get a good response. Similarly, we combine many of these features to understand if an image region contains a human face.

What are Integral Images?

In the previous section, we have seen that to calculate a value for each feature, we need to perform computations on all the pixels inside that particular feature. In reality, these calculations can be very intensive since the number of pixels would be much greater when we are dealing with a large feature.

The integral image plays its part in allowing us to perform these intensive calculations quickly so we can understand whether a feature or several features fit the criteria.

An integral image (also known as a summed-area table) is the name of both a data structure and an algorithm used to obtain this data structure. It is used as a quick and efficient way to calculate the sum of pixel values in an image or rectangular part of an image.

How is AdaBoost used in viola jones algorithm?

Next, we use a Machine Learning algorithm known as AdaBoost. But why do we even want an algorithm?

The number of features that are present in the 24x24 detector window is nearly 160,000, but only a few of these features are important to identify a face. So we use the AdaBoost algorithm to identify the best features in the 160,000 features.

In the Viola-Jones algorithm, each Haar-like feature represents a weak learner. To decide the type and size of a feature that goes into the final classifier, AdaBoost checks the performance of all classifiers that you supply to it.

To calculate the performance of a classifier, you evaluate it on all subregions of all the images used for training. Some subregions will produce a strong response in the classifier. Those will be classified as positives, meaning the classifier thinks it contains a human face. Subregions that don't provide a strong response don't contain a human face, in the classifiers opinion. They will be classified as negatives.

The classifiers that performed well are given higher importance or weight. The final result is a strong classifier, also called a boosted classifier, that contains the best performing weak classifiers.

So when we're training the AdaBoost to identify important features, we're feeding it information in the form of training data and subsequently training it to learn from the information to predict. So ultimately, the algorithm is setting a minimum threshold to determine whether something can be classified as a useful feature or not.

What are Cascading Classifiers?

Maybe the AdaBoost will finally select the best features around say 2500, but it is still a time-consuming process to calculate these features for each region. We have a 24x24 window which we slide over the input image, and we need to find if any of those regions contain the face. The job of the cascade is to quickly discard non-faces, and avoid wasting precious time and computations. Thus, achieving the speed necessary for real-time face detection.

We set up a cascaded system in which we divide the process of identifying a face into multiple stages. In the first stage, we have a classifier which is made up of our best features, in other words, in the first stage, the subregion passes through the best features such as the feature which identifies the nose bridge or the one that identifies the eyes. In the next stages, we have all the remaining features.

When an image subregion enters the cascade, it is evaluated by the first stage. If that stage evaluates the subregion as positive, meaning that it thinks it's a face, the output of the stage is maybe.

When a subregion gets a maybe, it is sent to the next stage of the cascade and the process continues as such till we reach the last stage.

If all classifiers approve the image, it is finally classified as a human face and is presented to the user as a detection.

Now how does it help us to increase our speed? Basically, If the first stage gives a negative evaluation, then the image is immediately discarded as not containing a human face. If it passes the first stage but fails the second stage, it is discarded as well. Basically, the image can get discarded at any stage of the classifier

3.3 Face Blurring in live video detection

Face blurring is a computer vision method used to anonymize faces in images and video.

Step 1: is to perform face detection.

Any face detector can be used here, provided that it can produce the bounding box coordinates of a face in an image or video stream.

Typical face detectors that you may use include

- Haar cascades
- HOG + Linear SVM
- Deep learning-based face detectors.

Step 2: is to extract the Region of Interest (ROI):

Your face detector will give you the bounding box (x, y)-coordinates of a face in an image. These coordinates typically represent:

- The starting x-coordinate of the face bounding box
- The ending x-coordinate of the face
- The starting y-coordinate of the face location
- The ending y-coordinate of the face
- You can then use this information to extract the face ROI itself

Step 3: is to actually blur

Typically, you'll apply a Gaussian blur to anonymize the face. You may also apply methods to pixelate the face if you find the end result more aesthetically pleasing.

Exactly how you "blur" the image is up to you — the important part is that the face is anonymized.

Using the original (x, y)-coordinates from the face detection (i.e., Step #2), we can take the blurred/anonymized face and then store it back in the original image (if you're utilizing OpenCV and Python, this step is performed using NumPy array slicing).

The face in the original image has been blurred and anonymized — at this point the face anonymization pipeline is complete.

3.4 Number plate detection

Automatic License/Number Plate Recognition (ANPR/ALPR) is a process involving the following steps:

Step 1: Detect and localize a license plate in an input image/frame

Step 2: Extract the characters from the license plate

Step 3: Apply some form of Optical Character Recognition (OCR) to recognize the extracted characters

ANPR tends to be an extremely challenging subfield of computer vision, due to the vast diversity and assortment of license plate types across states and countries.

License plate recognition systems are further complicated by:

- Dynamic lighting conditions including reflections, shadows, and blurring
- Fast-moving vehicles
- Obstructions

Additionally, large and robust ANPR datasets for training/testing are difficult to obtain due to:

- These datasets containing sensitive, personal information, including time and location of a vehicle and its driver
- ANPR companies and government entities closely guarding these datasets as proprietary information

Therefore, the first part of an ANPR project is usually to collect data and amass enough example plates under various conditions.

Creating new filters saving them in xml file and using those filters on to images

Unit 4: Computer Vision with OpenVINO

Learning Outcomes:

- Understand the concept of OpenVINO Toolkit
- Understand the Workflow of OpenVINO Toolkit
- ModelZoo & Model Optimizers use for OpenVINO-IR
- Able to create application on Edge Computing Using OpenVINO & Raspberry-PI

4.1 Introduction to OpenVINO



Reference : [gray and green computer processor and black motherboard photo – Free Image on Unsplash](#)

OpenVINO stands for Open Visual Inference and Neural Network Optimization. It is a toolkit provided by Intel to facilitate faster inference of deep learning models. It helps developers to create cost-effective and robust computer vision applications. It enables deep learning inference at the edge and supports heterogeneous execution across computer vision accelerators — CPU, GPU, Intel® Movidius™ Neural Compute Stick, and FPGA. It supports a large number of deep learning models out of the box. You can check out this link to know more about the model zoo.

OpenVINO is a cross-platform deep learning toolkit developed by Intel. The name stands for “Open Visual Inference and Neural Network Optimization.” OpenVINO

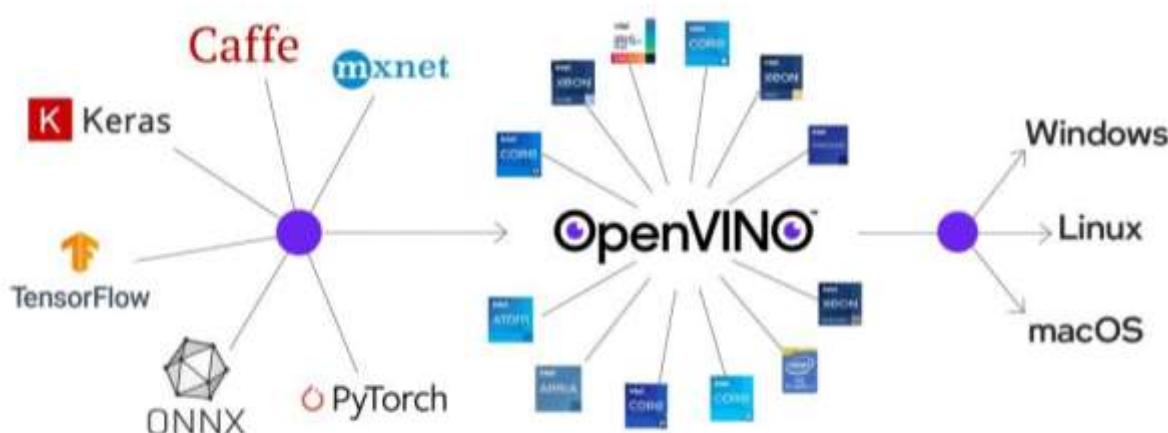
focuses on optimizing neural network inference with a write-once, deploy-anywhere approach for Intel hardware platforms.

The toolkit is free for use under Apache License version 2.0 and has two versions:

- OpenVINO toolkit, which is supported by the open-source community and the
- Intel Distribution of OpenVINO toolkit, which is supported by Intel.

Using the OpenVINO toolkit, software developers can select models and deploy pre-trained deep learning models (YOLO v3, ResNet 50, etc.) through a high-level C++ Inference Engine API integrated with application logic.

Hence, OpenVINO offers integrated functionalities for expediting the development of applications and solutions that solve several tasks using computer vision, automatic speech recognition, natural language processing, recommendation systems, machine learning, and more.



Reference: [What is OpenVINO? - The Ultimate Overview \(Updated\) - viso.ai](https://www.viso.ai/what-is-openvino-the-ultimate-overview-updated/)

Deep Neural Networks (DNNs) have made considerable advances in many industrial domains in the past few years, bringing the accuracy of computer vision algorithms to a new level. However, deploying and producing such accurate and useful models requires adaptations for the hardware and computational methods.

4.2 OpenVINO Toolkit Components

The two main components of the OpenVINO toolkit are Model Optimizer and Inference Engine. So, we will dive into their details, to better understand their role and internal working.

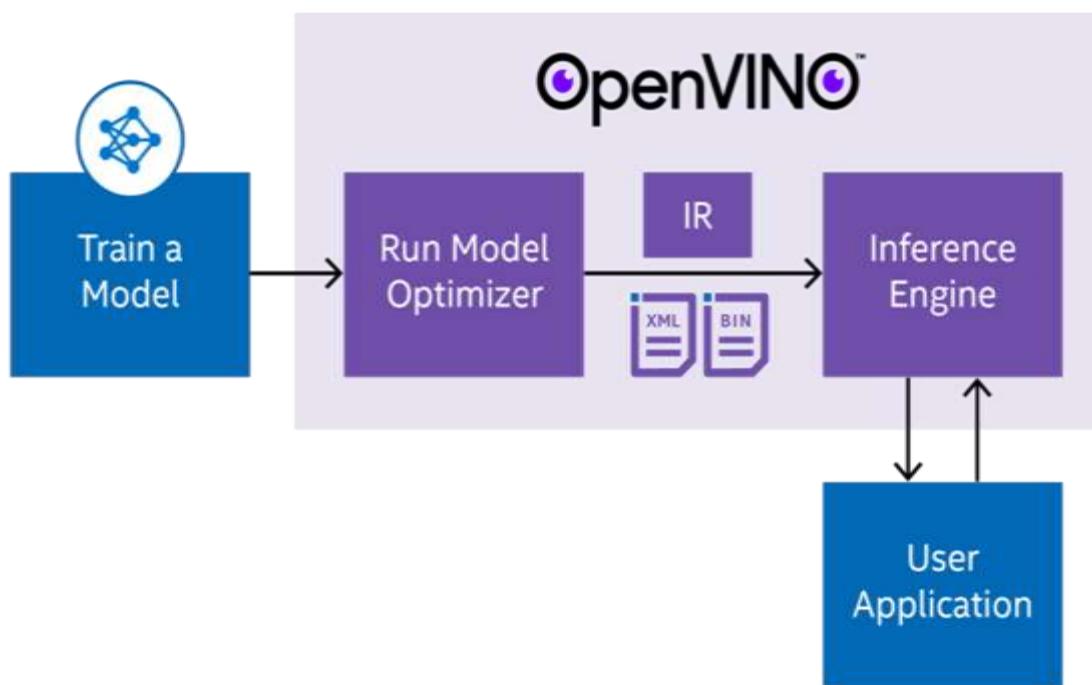
4.3 Model Optimizer

Model optimizer is a cross-platform command-line tool that facilitates the transition between the training and deployment environment. It adjusts the deep learning models for optimal execution on end-point target devices.

Working

Model Optimizer loads a model into memory, reads it, builds the internal representation of the model, optimizes it, and produces the **Intermediate Representation**. Intermediate Representation is the only format that the Inference Engine accepts and understands. The Model Optimizer does not infer models. It is an offline tool that runs before the inference takes place. Model Optimizer has two main purposes:

- **Produce a valid Intermediate Representation.** The primary responsibility of the Model Optimizer is to produce two files (.xml and .bin) that form the Intermediate Representation.
- **Produce an optimized Intermediate Representation.** Pretrained models contain layers that are important for training, such as the Dropout layer. These layers are useless during inference and might increase the inference time. In many cases, these layers can be automatically removed from the resulting Intermediate Representation. However, if a group of layers can be represented



- As one mathematical operation, and thus as a single layer, the Model Optimizer recognizes such patterns and replaces these layers with only one. The result is an Intermediate Representation that has fewer layers than the original model. This decreases the inference time.

Operations of Model Optimizer

Reshaping: The Model Optimizer allows us to reshape our input images. Suppose you have trained your model with an image size of 256 * 256 and you want to convert the image size to 100 * 100, then you can simply pass on the new image size as a command-line argument and the Model Optimizer will handle the rest for you.

Batching: We can change the batch size of our model at inference time. We can just pass the value of batch size as a command-line argument. We can also pass our image size like this [4,3,100,100]. Here we are specifying that we need 4 images with dimensions 100*100*3 i.e RGB images having 3 channels and having width and height as 100. Important thing to note here is that now the inference will be slower as we are using a batch of 4 images for inference rather than using just a single image.

Modifying the Network Structure: We can modify the structure of our network, i.e we can remove layers from the top or from the bottom. We can specify a particular layer from where we want the execution to begin or where we want the execution to end.

Standardizing and Scaling: We can perform operations like normalization (mean subtraction) and standardization on our input data.

Quantization: It is an important step in the optimization process. Most deep learning models generally use the FP32 format for their input data. The FP32 format consumes a lot of memory and hence increases the inference time. So, intuitively we may think, that we can reduce our inference time by changing the format of our input data. There are various other formats like FP16 and INT8 which we can use, but we need to be careful while performing quantization as it can also result in loss of accuracy. Using the INT8 format can help us in reducing our inference time significantly, but currently, only certain layers are compatible with the INT8 format: Convolution, ReLU, Pooling, Eltwise and Concat. So, we essentially perform hybrid execution where some layers use FP32 format whereas some layers use INT8 format. There is a separate layer that handles these conversions. i.e we don't have to explicitly specify the type conversion from one layer to another.

Calibrate layer: handles all these intricate type conversions. The way it works is as follows:

- Initially, we need to define a threshold value. It determines the drop in accuracy we are willing to accept.
- The Calibrate layer then takes a subset of data and tries to convert the data format of layers from FP32 to INT8 or FP16.
- It then checks the accuracy drop and if it less than the specified threshold value, then the conversion takes place.

Inference Engine

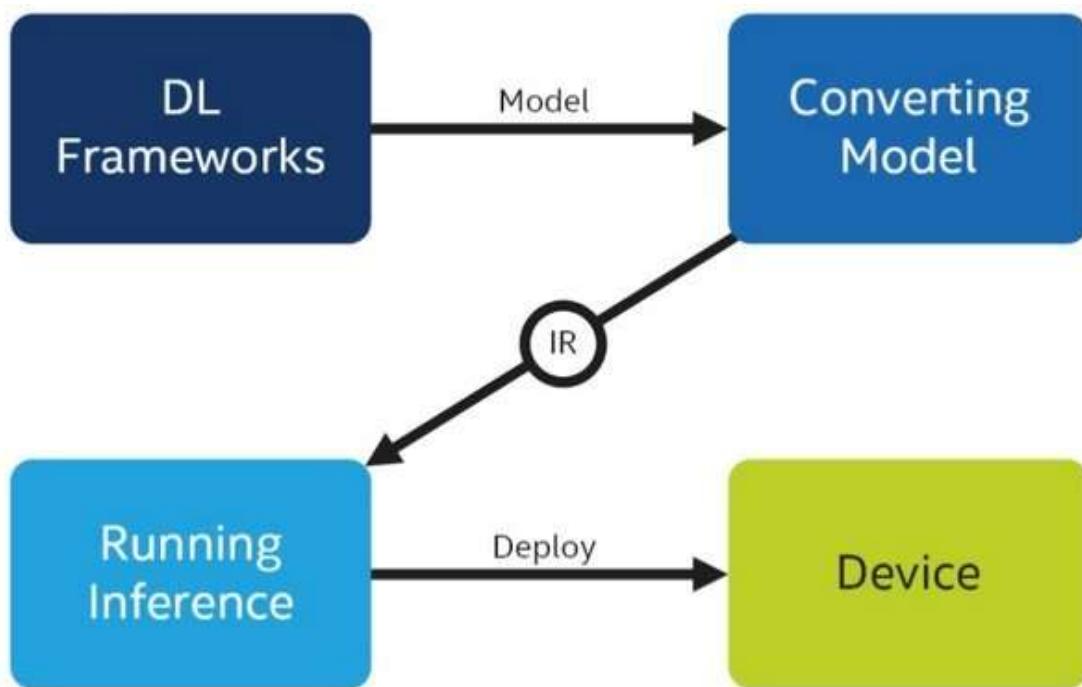
After using the Model Optimizer to create an intermediate representation (IR), we use the Inference Engine to infer input data. The Inference Engine is a C++ library with a set of C++ classes to infer input data (images) and get a result. The C++ library provides an API to read the Intermediate Representation, set the input and output formats, and execute the model on devices. The heterogeneous execution of the model is possible because of the Inference Engine. It uses different plug-ins for different devices.

Heterogeneous Plug-in: We can execute the same program on multiple devices. We just need to pass in the target device as a command-line argument and the Inference Engine will take care of the rest i.e. we can run the same piece of code on a CPU, GPU, VPU or any other device compatible with the OpenVINO toolkit.

- We can also execute parts of our program on different devices i.e. some part of our program might run on CPU and other parts might be running on a FPGA or a GPU. If we specify HETERO: FPGA, CPU then the code will run primarily on an FPGA, but if suppose it encounters a particular operation which is not compatible with FPGA then it will switch to CPU.
- We can also execute certain layers on a specific device. Suppose you want to run the Convolution layer only on your GPU then you can explicitly specify it.
- The important thing to note here is that we need to be careful about the data format while specifying different hardware. Not all devices work with all the data types. Example — The Neural Compute Stick NCS2, which comes with a Movidius chip, doesn't support the INT8 format.

OpenVINO allows the optimization of DNN models for inference to be a streamlined, efficient process through the integration of various tools. The OpenVINO toolkit is based on the latest generations of Artificial Neural Networks (ANN), such as Convolutional Neural Networks (CNN) as well as recurrent and attention-based networks.

The OpenVINO toolkit covers both computer vision and non-computer vision workloads across Intel hardware. It maximizes performance and accelerates application development. OpenVINO aims to accelerate AI workloads and speed up time to market using a library of predetermined functions as well as pre-optimized kernels. In addition, other computer vision tools such as OpenCV, OpenCL kernels, and more are included in the OpenVINO toolkit.



Reference: [OpenVINO™ Documentation — OpenVINO™ documentation — Version\(latest\)](#)

4.4 Benefits of OpenVINO

- **Accelerate Performance:** Expedite computer vision workloads by enabling simple execution methods across different Intel processors and accelerators such as CPU, GPU/Intel Processor Graphics, VPU (Intel AI Stick NCS2 with Myriad X), and FPGA.
- **Streamline Deep Learning Deployment:** Utilize Convolutional Neural Network (CNN)-based deep learning functions using one common API in addition to more than 30 pre-trained models and documented code samples. With more than 100 public and custom models, the OpenVINO toolkit streamlines deep learning innovation by providing one centralized method for implementing dozens of deep learning models.
- **Extend and Customize:** OpenCL (Open Computing Language) Kernels and other tools offer an open, royalty-free standard way to add custom code pieces straight into the workload pipeline, customize deep learning model layers without the burden of framework overheads, and implement parallel programming of various accelerators.
- **Innovate Artificial Intelligence:** The complete Deep Learning Deployment Toolkit within OpenVINO allows users to extend artificial intelligence within private applications and optimize artificial intelligence “all the way to the cloud” with processes such as the Model Optimizer, Intermediate Representation, nGraph Integration, and more.
- **Full Viso Suite Integration (End-To-End):** OpenVINO is fully integrated with the enterprise no-code computer vision platform Viso Suite. Viso Suite provides

pre-built modules to fetch the video feed of any digital camera (IP cameras, webcams, etc.) and multi-camera support. Visual programming with logic workflows allows fast building and updating of complete computer vision applications that can be deployed to edge devices – all within one platform.

4.5 Practical: Installing Intel OpenVINO Toolkit (Linux)

In this section, we will cover the installation steps for the Ubuntu 20.04 OS, 2021.3 version (the latest version available currently) of the Intel OpenVINO Toolkit. The same steps also apply, if you are on Ubuntu 18.04. We've listed each step to install Intel OpenVINO Toolkit in detail here. Let's go over it together now.

Step 1: Download the Correct Version of OpenVINO Toolkit

Head over to the official download page to get the correct version of OpenVINO, after choosing all the prerequisites according to your needs and OS.

The screenshot shows the 'Choose a Preferred Package' section of the Intel OpenVINO Toolkit download page. It includes fields for Environment (Dev Tools selected), Operating System (Linux selected), OpenVINO™ Version (2021.4.2 LTS selected), Language (Python selected), and Distribution (Offline Installer selected). A 'Download' button is at the bottom.

Environment	Dev Tools	Runtime
Best option to develop and optimize deep-learning models		You already have a model and want to run inference on it

Operating System	Windows	macOS	Linux
			Selected

OpenVINO™ Version	2022.1 (recommended)	2021.4.2 LTS	2020.3 LTS
Latest standard release		Selected (Latest LTS release)	Previous LTS release

Language	Python	C++
Included by default, and cannot be unselected		

Distribution	Offline Installer Recommended option	Online Installer	PIP
APT (Requires Ubuntu Linux)		YUM (Requires Red Hat Linux)	GitHub Source
Gitee Source		Docker	

[Learn more about distribution options](#)
[Try OpenVINO on Intel® DevCloud](#)

Download Intel® Distribution of OpenVINO™ Toolkit

[Install instructions](#)
[Get started guide](#)
[OpenVINO Notebooks](#)

[Download](#)

Fig: Download page

For example, the following image shows us selecting to download the local installer of OpenVINO 2021.3 version for the Linux operating system. Intel OpenVINO Toolkit official download page. Choosing the correct version of OpenVINO Toolkit, as per the operating system.

Which version should you download? Any of the older, available versions will work too. But better to go with the latest version for it provides the most updated functionalities of the toolkit. Choose any version, starting from OpenVINO 2020. Just keep in mind that all these versions are fully supported only on Ubuntu 18.04 LTS and 20.04 LTS.

Click the download button. It will download a file named `I_openvino_toolkit_p_<version>.tgz`, where `<version>` is the version number you have chosen to download.

Step 2: Unpack the Installer File

Open your terminal and cd into the directory where you kept the OpenVINO installer file. Then unpack the file, using the following command:

```
1 | tar -xvzf I_openvino_toolkit_p_<version>.tgz
```

After unpacking, you will see a `I_openvino_toolkit_p_<version>` folder in your current working directory. This contains the OpenVINO installer and other required files.

Step 3: Install OpenVINO Toolkit

Now, cd into the `I_openvino_toolkit_p_<version>`. You get three options to install OpenVINO on your system.

1. The graphical user interface (GUI) installation wizard
2. The command line installer
3. The command line installer, with silent instructions.

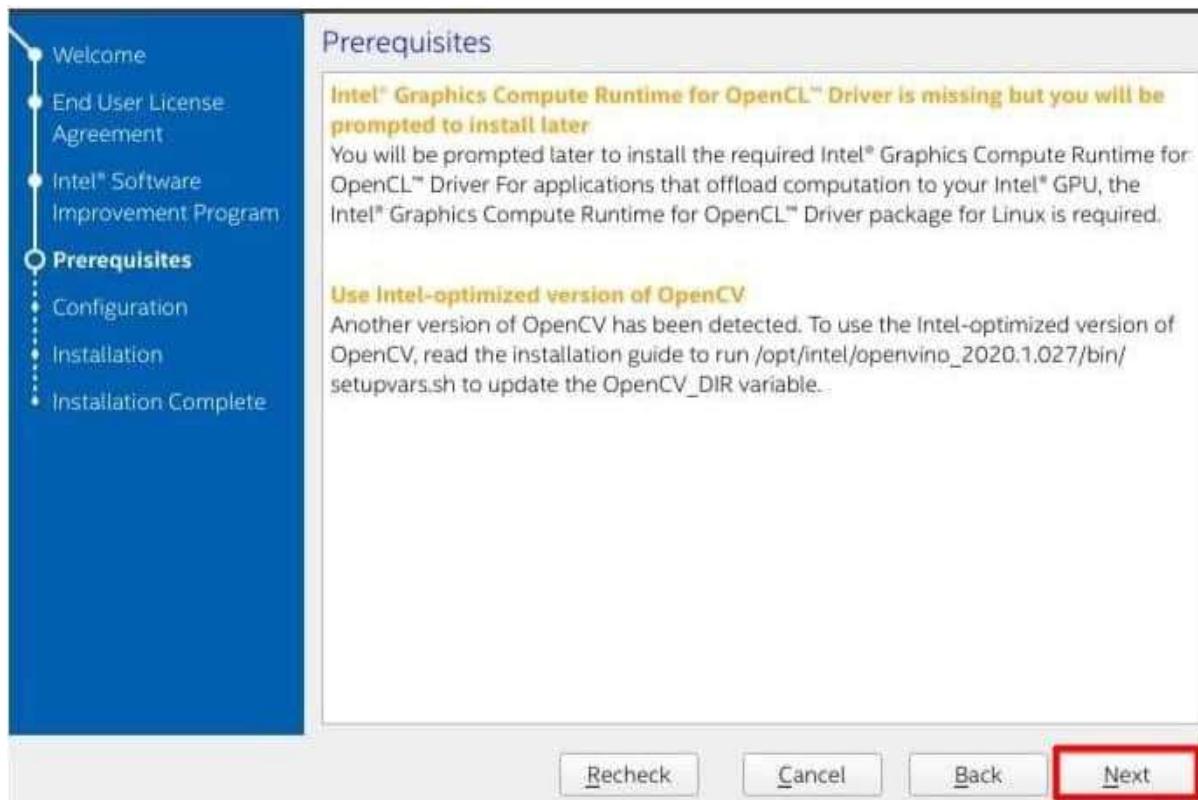
We will use the GUI installation wizard, that is the `install_GUI.sh` file, as it is the easiest and most intuitive to follow. To start the installation, type the following command in your terminal.

```
1 | sudo ./install_GUI.sh
```

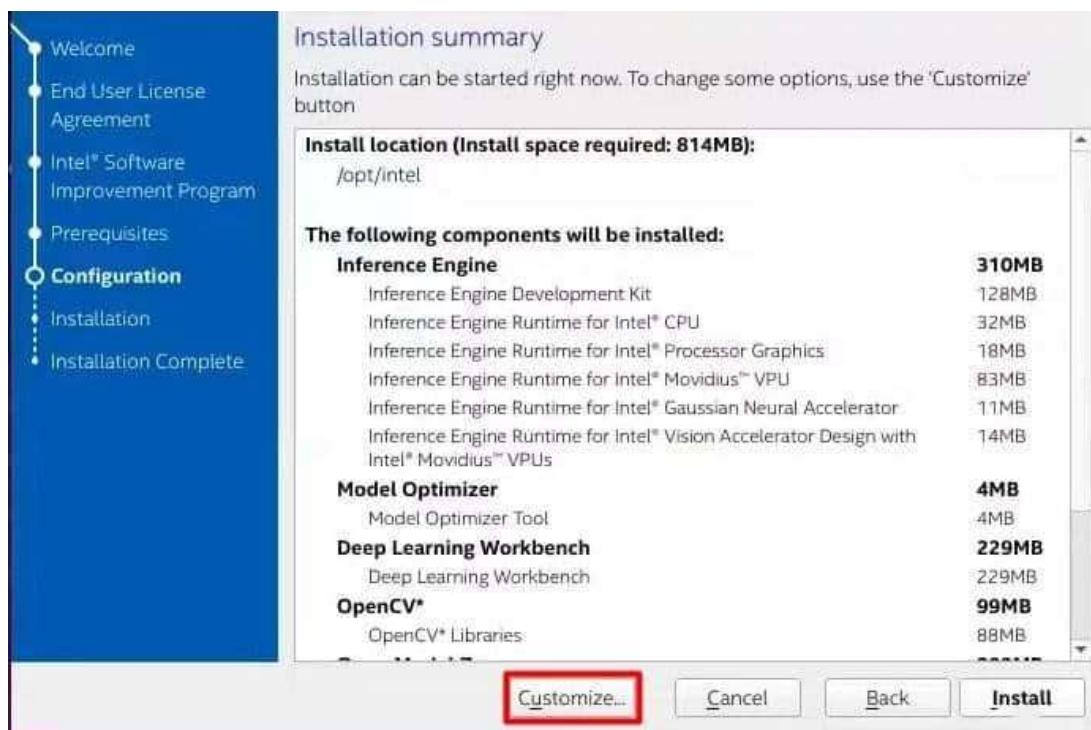
Note: You can also continue with the installation processes, without the **sudo** access.

Step 4: Following the Installation Instruction on Screen

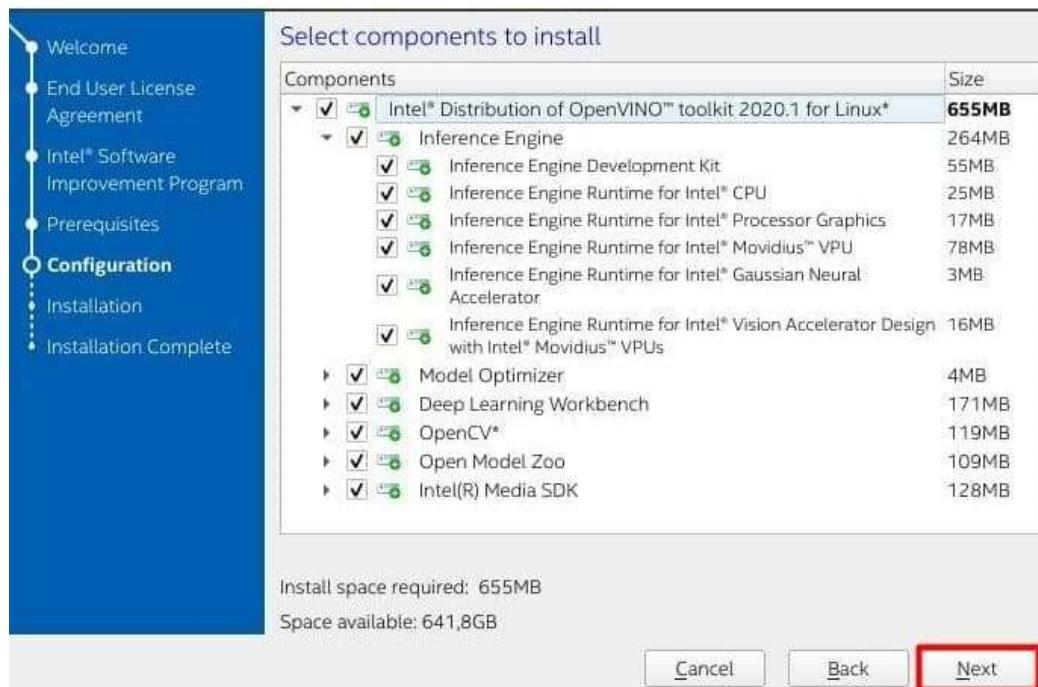
First, comes the welcome screen, and then the license agreement. Next, you will see the prerequisites screen, similar to the image below.



Intel OpenVINO prerequisites installation page shown above. Even if you get warnings for not meeting the prerequisites, you can still proceed with the installation. These can be installed later, when you are installing the dependencies. After this comes the installation-configuration summary screen.



Configuring the Intel OpenVINO installation. If you wish, you can also customize the configurations, and choose what you want to install.



Selecting the desired configuration, while installing OpenVINO Toolkit. First, choose the components, then move forward with the installation. It takes some time to complete the installation, and you will see a screen, as in the image below:



Finish the Intel OpenVINO Installation. If you installed OpenVINO Toolkit with administrator privileges, in line with the steps listed above, then it will be installed in the `/opt/intel/openvino_<version>/` directory.

The next few steps ensure you face no issues while using this toolkit. These steps are necessary irrespective of you wanting to use any advanced functionality of OpenVINO or not. They are also needed to properly run the models available in the Intel Model Zoo.

Step 5: Installing Software Dependencies

These include software dependencies for:

1. Intel-optimized build of OpenCV library
2. Deep Learning Inference Engine
3. Deep Learning Model-Optimizer tools

Change your current working directory to the `install_dependencies` folder.

```
1 cd /opt/intel/openvino_2021/install_dependencies
```

Run the following script to install the dependencies.

```
1 sudo -E ./install_openvino_dependencies.sh
```

Step 6: Set Up the Environment Variables

To set the environment variables, open a new terminal and type:

```
1 | vi ~/.bashrc
```

Go to the end of the file, and add the following line:

```
1 | source /opt/intel/openvino_2021/bin/setupvars.sh
```

Note: If you have installed the OpenVINO Toolkit in any other directory of your choice, then please provide that path. In such cases, it should be <path_to_your_directory>/openvino_2021/bin/setupvars.sh.

Now, save and close the file. Shut down your current terminal and open a new one so that system-wide changes can take place. We're only one step away from completing installation now. Just configure the Model Optimizer and you're done.

Step 7: Installing Model Optimizer Prerequisites

The Model Optimizer is a command-line tool in the OpenVINO Toolkit. Use this Model Optimizer to convert models, trained with different frameworks, into a format accepted by the OpenVINO Toolkit for inference. For your information, the OpenVINO Toolkit does not support inference directly. None of the models trained with Deep-Learning frameworks like TensorFlow, Caffe, MXNet, ONNX, or even Kaldi can help you infer. First, you'll need to convert these trained models into an Intermediate Representation (IR), which consists of:

1. A .xml file, which describes the network architecture
2. A .bin file that holds the weights and biases of the trained model

To convert, run the models through the Model Optimizer, after installing the necessary prerequisites. To configure the Model Optimizer, open your terminal and go to the Model Optimizer prerequisites directory.

```
1 | cd opt/intel/openvino_2021/deployment_tools/model_optimizer/install_prerequisites
```

Execute the following command to run the script that will configure the Model Optimizer for Caffe, TensorFlow, MXNet, ONNX and even Kaldi, at one go.

```
1 | sudo ./install_prerequisites.sh
```

This completes the installation process for the OpenVINO Toolkit. Now, you are all set to use any model from the Intel Model Zoo, or to convert the models for inference.

4.6 Practical: Installing Intel OpenVINO Toolkit (Windows)

These are the steps of configuring the Intel OpenVINO toolkit runtime inference engine on windows platform. Kindly ensure the system should have python 3.9(not any other version) and VS code installed.

Step 1: Download and Install OpenVINO Core Components¶

1. Open a new command prompt window as administrator by right-clicking Command Prompt from the Start menu and select Run as administrator, and then run the following command:

```
mkdir "C:\Program Files (x86)\Intel"
```

2. Go to the Downloads folder or any other path where you want to download the zip file of OpenVINO toolkit. And run the following command:

```
curl https://storage.openvinotoolkit.org/repositories/openvino/packages/2022.2/windows/w_openvino_toolkit_windows_2022.2.0.7713.af16ea1d79a_x86_64.zip --output openvino_2022.2.0.7713.zip -L
```

3. Run the following commands for extracting the file in Intel folder:

```
tar -xf openvino_2022.2.0.7713.zip
```

```
ren w_openvino_toolkit_windows_2022.2.0.7713.af16ea1d79a_x86_64 openvino_2022.2.0.7713
```

```
move openvino_2022.2.0.7713 "C:\Program Files (x86)\Intel"
```

Step 2: Configure the Environment¶

1. Setup the environment to openvino

```
"C:\Program Files (x86)\Intel\openvino_2022\setupvars.bat"
```

2. Installing OpenVINO™ Development Tools¶

Set Up Python Virtual Environment¶

```
python -m venv openvino_env
```

3. Activate Virtual Environment¶

```
openvino_env\Scripts\activate
```

4. Set Up and Update PIP to the Highest Version¶

```
python -m pip install --upgrade pip
```

5. Install the Package¶

For example, to install and configure the components for working with TensorFlow 2.x and ONNX, use the following command:

```
pip install openvino-dev[tensorflow2,onnx]
```

6. Test the Installation¶

To verify the package is properly installed, run the command below (this may take a few seconds):

```
mo -h
```

Now downloading the predefined models in OpenVINO notebook:

- **Install Python**

Download 64 bit version of Python software (3.9) from [python.org](https://www.python.org/).

Run the installer by double clicking it. Follow the installation steps to set up the software.

While installing, make sure you check the box to *add Python to system PATH*.

Note

Python software available in the Microsoft Store is not recommended. It may require additional packages.

- **Install GIT**

Download 64 bit version of GIT from git-scm.org

Run the installer by double clicking it. Follow the installation steps to set up the software.

- **Install C++ Redistributable (For Python 3.8 only)**

Download 64 bit version of C++ Redistributable from [here](#)

Run the installer by double clicking it. Follow the installation steps to set up the software.

Installing notebooks

Clone the Repository

Using the --depth=1 option for git clone reduces download size.

```
git clone --depth=1 https://github.com/openvinotoolkit/openvino_notebooks.git
```

```
cd openvino_notebooks
```

Upgrade PIP

```
python -m pip install --upgrade pip
```

Install required packages

```
pip install -r requirements.txt
```

Install the virtualenv Kernel in Jupyter

```
python -m ipykernel install --user --name openvino_env
```

Run the Notebooks

Launch a Single Notebook

If you want to launch only one notebook, such as the *Monodepth* notebook, run the command below.

```
jupyter openvino_notebooks\notebooks\201-vision-monodepth\201-vision-monodepth.ipynb
```

Note: If you get an error no command named Jupyter then install Jupyter using the following command.

Pip install Jupyter

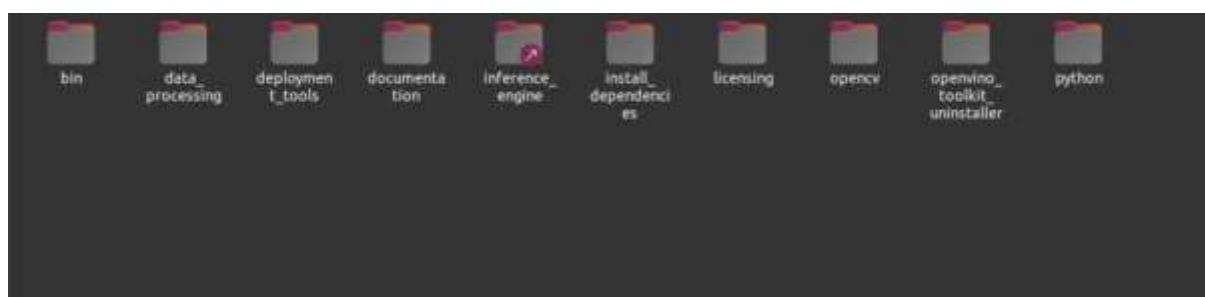
Pip install jupyterlab

Launch All Notebooks

jupyter lab notebooks

4.7 Exploring OpenVINO Toolkit Directories

Navigating through the freshly installed OpenVINO Toolkit can be a bit overwhelming for newcomers. It contains a lot of directories, with each directory having numerous sub-directories.

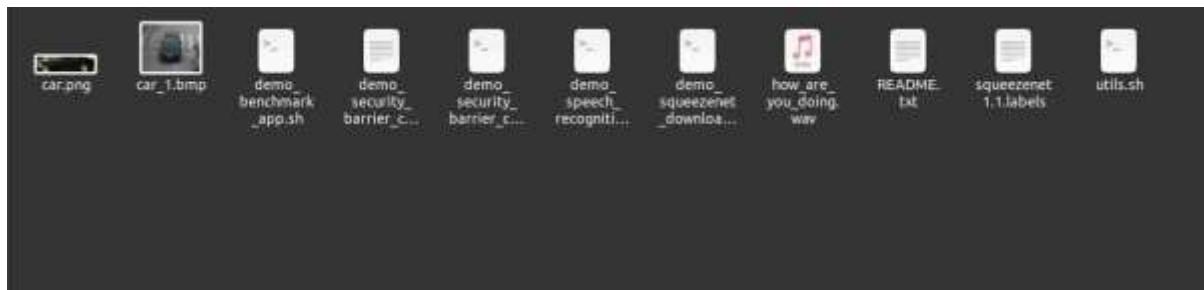


Look at the image above. You will see a similar structure inside your installed directory. We are going to discuss the directories and contents present in OpenVINO installed

super directory. As you can refer to installed folder, in total the OpenVINO directory contains approx 1266 sub directories and 6032 files. Let's go through the important ones now. While using the OpenVINO Toolkit, for most of our work, we'll be referring to the deployment_tools directory. So, let's explore the sub directories in this directory first.

The demo Directory

The demo directory contains some demos that you can run directly, using off-the-shelf optimized models. These models will download automatically when you execute their respective scripts. Your directory structure should look, as shown below:



The open_model_zoo Directory

This is one of the most important directories, so you better get comfortable with it. You'll find here a list of all the official and public models available for use with the OpenVINO Toolkit. Along with that, there are utility scripts to download these models. The following image shows the sub directories and files inside the open_model_zoo directory.



Going over some of its subdirectories:

- The demos directory contains a lot of demo codes that we can execute using the OpenVINO Toolkit, after downloading the models from the Model Zoo. These range from classification to action recognition to object detection, and many more. This is one directory you must definitely explore.
- Coming to the models directory, it contains two subdirectories:

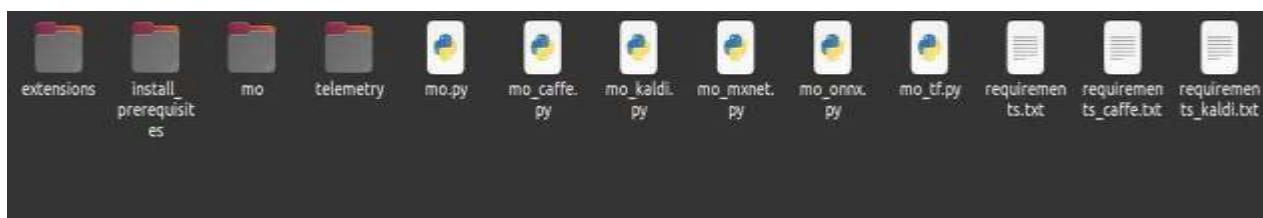
1. intel – The intel folder contains a list of all the official pre-models provided by Intel. These models get downloaded directly in the FP32, FP16, and INT8 precision format, and need no further conversion.
2. public – The models from this folder however are downloaded in the format of the Deep Learning framework they were trained on. For example, TensorFlow models will have .pb format, Caffe models will have .caffemodel, and so on.

Another thing to note here is that these model folders do not contain the actual model weight files. They only have the model documentation and some .yml configuration files. The documentation is a Markdown file containing the model benchmark results, the original framework, parameters, along with information on the input and output format. To download one of the intel or public models, you need the downloader.py script, which is present in the tools/downloader subdirectory. While executing the script, you will have to provide the name of the model you want to download. Either give the exact folder name from the models directory, or get the model name from the Markdown documentation file. More on this, when we execute one of the demos.

Another important subdirectory is the accuracy_checker inside tools. For now, just know that we can check the accuracy of different models, using the scripts in this directory.

The model_optimizer Directory

The model_optimizer also happens to be a very important directory, so understand it well. It contains the scripts that convert the trained neural network models to the Intermediate Representation (.xml and .bin files) that OpenVINO accepts. This is mainly for the public models which are not downloaded in the IR format by default.



There are separate Python scripts to convert models from different frameworks like TensorFlow, Caffe, ONNX and MXNet to the OpenVINO IR format. These scripts are named as follows:

- mo_tf.py for TensorFlow models
- mo_caffe.py for Caffe models
- mo_onnx.py for ONNX models
- mo_mxnet.py for MXNet models

Along with these, there is also one **mo.py** which acts like a universal model converter script. You can just use this to convert any trained model into the OpenVINO IR format. In next coming section, we will also show you how to use the Model Optimizer, and the way models are converted from Caffe and TensorFlow frameworks to the OpenVINO IR format.

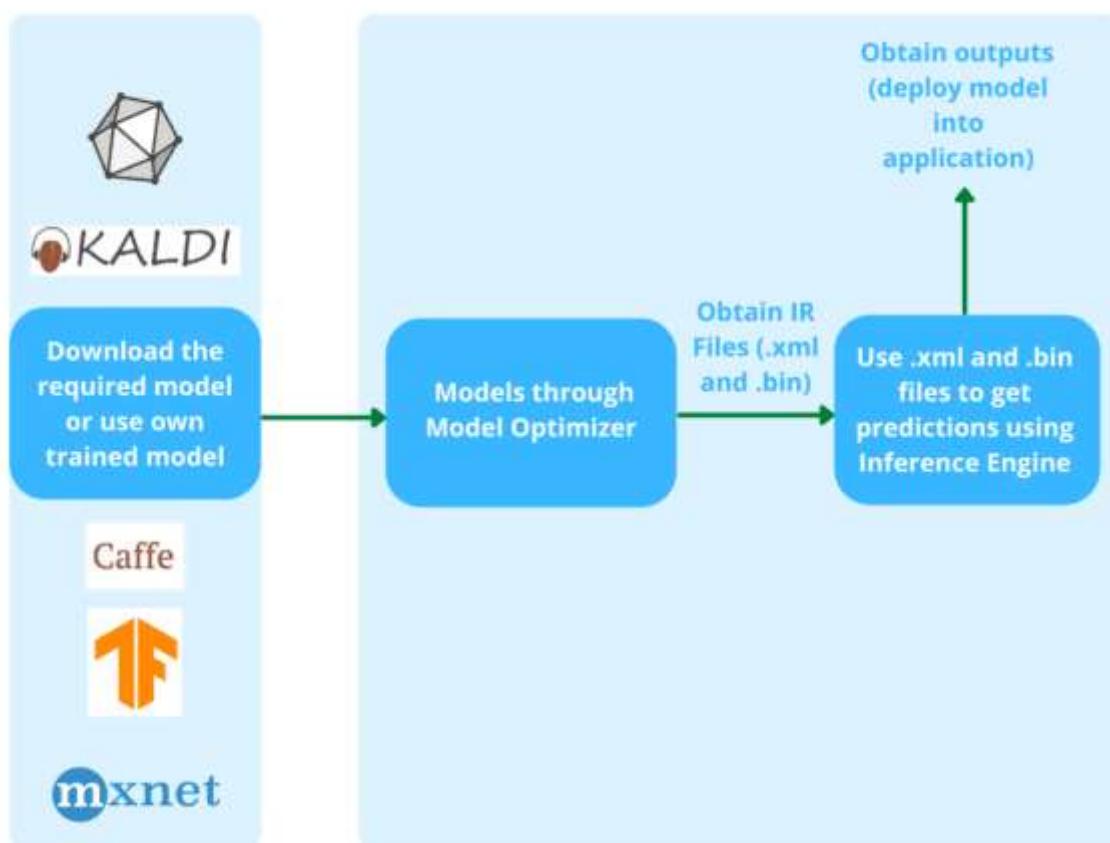
4.8 Working with Model Optimizer

Let's now focus on the model-conversion process. Here, you will basically learn to convert pre-trained Deep Learning models from supporting frameworks to the OpenVINO IR format. Well, these are the frameworks that OpenVINO toolkit supports: Caffe, TensorFlow, MXNet, Kaldi, PyTorch and ONNX.

Specifically, this section will cover the conversion of models from:

- Caffe to OpenVINO IR format

The basic conversion workflow of a pre-trained model from a specific framework to the OpenVINO Toolkit format looks like this:



Reference: [Model Optimizer](#)

The basic conversion workflow of a pre-trained model into the OpenVINO format. Obviously, there is more to the conversion process. We will cover those details in

coming sections, where we will be talking about IR files generation from deep learning model

Converting a SqueezeNet Caffe Model to OpenVINO-IR Format

Let's start with an image classification model, that is the **SqueezeNet Caffe model**. It is one of the publicly available models from Model Zoo.

Start the process by following these simple steps:

- Download the SqueezeNet Caffe model from the public Model Zoo.
- Run the model optimizer to convert the Caffe model to IR format.

Begin by downloading the SqueezeNet-Caffe model. Our model is named `squeezezenet1.1` (a pretrained deep learning mode) in the public subdirectory, inside the `open_model_zoo` directory. Just provide the model name, while executing the `downloader.py` script to ensure the correct model is downloaded. Now head over to the `tools/downloader` directory, inside the `deployment_tools`. Go there directly by using this specific command:

```
1 | cd /opt/intel/openvino_2021/deployment_tools/open_model_zoo/tools/downloader
```

Here, we execute the `downloader.py` script, with the following command:

```
1 | python3 downloader.py --name squeezezenet1.1
```

Note: The `--name` flag accepts the exact name of the model we want to download. Giving a model name that is neither in intel nor in the public model directory will surely result in an error.

Post execution, you will see a public folder in the current working directory, containing the `squeezezenet1.1` sub directory. It will consist of three files:

- `squeezezenet1.1.caffemodel`
- `squeezezenet1.1.prototxt`
- `squeezezenet1.1.prototxt.orig`

Out of these, the ones that interest us the most are the `squeezezenet1.1.caffemodel` and `squeezezenet1.1.prototxt` files. We need these to run the Model Optimizer and obtain the `.xml` and `.bin` files. Details of individual file is listed below:

- The .caffemodel file contains the model weights.
- And the .prototxt file contains the model architecture required by the Model Optimizer.

Next, we run the model optimizer and convert the Caffe model into IR format. So, head over to the model_optimizer directory.

```
1 | cd /opt/intel/openvino_2021_3_latest/openvino_2021/deployment_tools/model_optimizer
```

Next, execute the following command:

```
1 | python mo.py --input_model squeezenet1.1.caffemodel --batch 1 --output_dir squeezenet_ir
```

Let us go over the flags we have used:

- **--input_model:** This is the path to the Caffe model that we want to convert into IR format. In the above example, we assume the Caffe model is present in the same directory as the mo.py script. Please note that even the squeezenet1.1.prototxt file should be present in the same directory, so that the script can infer the path to the file on its own. Else you will have to provide the path to the .prototxt file, using the **--input_proto** flag.
- **--batch:** This flag specifies the batch size for building the OpenVINO models. It comes into play during inference. By default, the batch size is 1. It is the batch size that determines the number of images/frames the model will infer on, when executing the inference scripts.
- **--output_dir:** This is the output directory in which the resulting .xml and .bin files will be stored. In the above example, we have provided it as squeezenet_ir. If absent, the directory will be automatically created.

If everything runs successfully, you should see an output similar to the one below:

Note that install_prerequisites scripts may install additional components.

[SUCCESS] Generated IR version 10 model.

[SUCCESS] XML file: /home/divakar/my_data/Data_Science/Projects/openvino_experiments/squeezenet1.1_caffemodel/squeezenet_ir/squeezenet1.1.xml

[SUCCESS] BIN file: /home/divakar/my_data/Data_Science/Projects/openvino_experiments/squeezenet1.1_caffemodel/squeezenet_ir/squeezenet1.1.bin

[SUCCESS] Total execution time: 6.39 seconds.

[SUCCESS] Memory consumed: 344 MB.

Inside the squeezenet_ir directory, you will find the two files we need:

- **squeezenet1.1.bin:** This contains the model weights.

- squeezenet1.1.xml: It has the model topology/architecture.

We have successfully converted our first image classification Caffe model to the appropriate IR format, which we can now leverage to run inference on Intel hardware.

4.9 OpenVINO Deep Learning Workbench

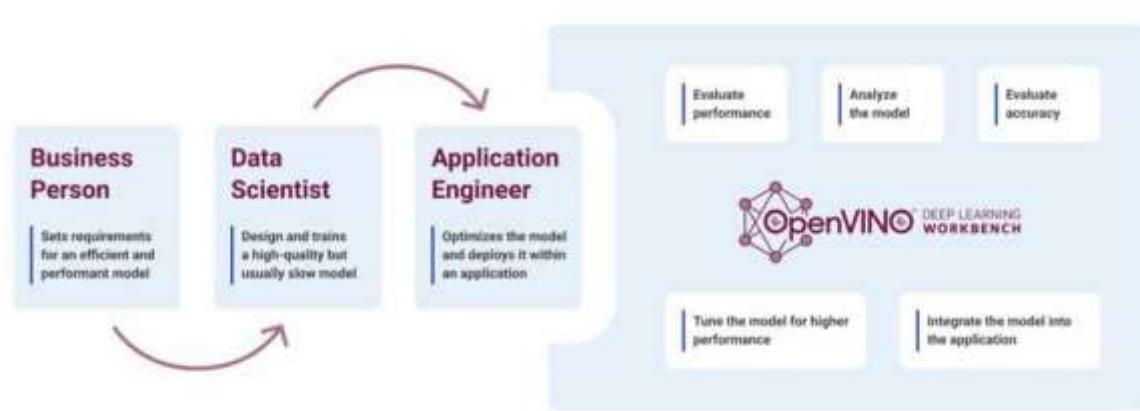
The Intel-OpenVINO Toolkit provides many great functionalities for Deep-Learning model optimization, inference and deployment. Perhaps the most interesting and practical tool among them is the Deep-Learning (DL) workbench. Not only does model optimization, calibration, and quantization get easier, but the OpenVINO-Deep Learning Workbench also makes the final model deployment-ready in a matter of minutes.

Let's understand what exactly is the DL workbench and why it's important. It is a web-based application provided by the Intel-OpenVINO toolkit that essentially runs in the browser. And its goal is to minimize the inference-to-deployment workflow timing for Deep-Learning models.

The DL workbench strongly integrates many of the optimization, quantization and deployment processes that OpenVINO supports, but are done manually. Its easy-to-use Graphical-User Interface lets you access almost everything, no need to bother what's going on below the hood. You can not only import models and datasets, but also visualize and optimize these models. Even compare accuracies across various runs and parameters. Also, you can export the final model, which will be deployment-ready.

Functionalities and Components of the DL Workbench

Let's study the functionalities and components that make the DL workbench so special.



OpenVINO Deep Learning Workbench

The general workflow of the DL workbench, showing the basic functionalities is shown in above figure. The general workflow of the DL workbench and know the basic functionalities that come integrated with the following components:

- Model Evaluator
- Model Optimizer
- Model Quantization Tool
- Accuracy Checker
- Deployment Package Manager

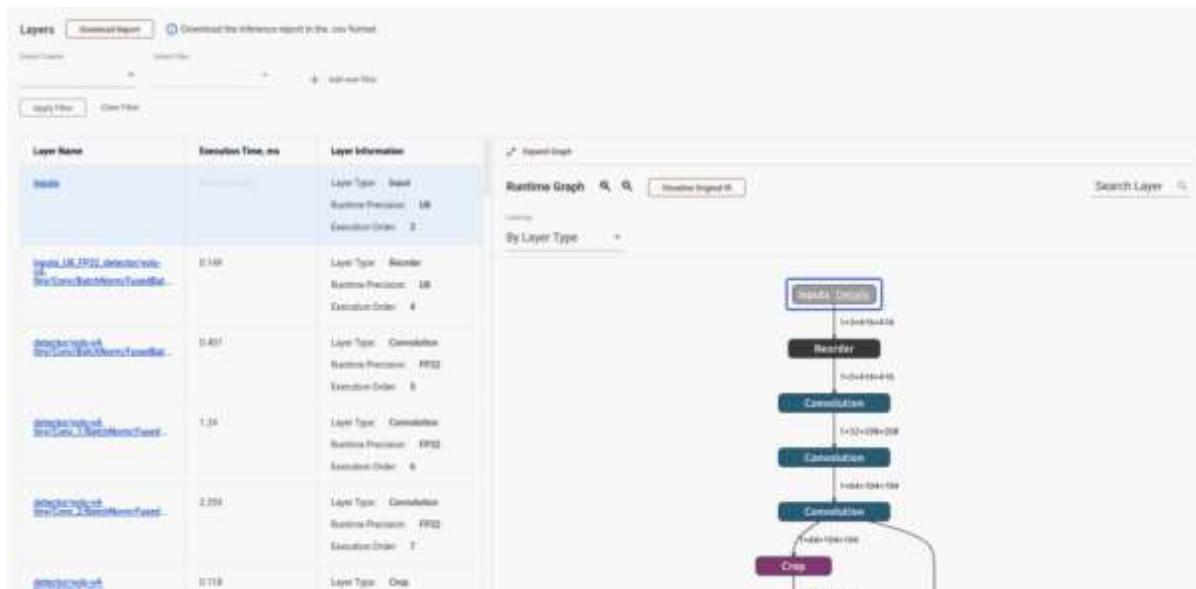
Now, let's go over them in a bit more detail.

1. Evaluating Model Performance

The DL workbench allows you to import any model from its list of supported frameworks, which includes TensorFlow, ONNX, Caffe, MXNet and Kaldi. But did you know you can evaluate their performance too? Also, you can simply import a dataset of your choice, like the MS COCO or the PASCAL VOC dataset, and run an evaluation on them. Even if you do not have a standardized dataset at hand, you can always generate random data in the DL workbench itself. Not only that, along with models from various frameworks, you can also import and evaluate models that are already in the OpenVINO-IR format.

2. Analyzing the Model

Furthermore, you get a great environment and visualization tools to analyze the architecture of your imported models.



[Model Architecture](#)

You can check each layer, each operation in the layers, and even how much time each layer took for every single operation. Then use these insights to further optimize your model and make it all the better.

3. Accuracy Evaluation

You can evaluate the accuracy of your imported models on various datasets, and this perhaps is the most important functionality available in the DL workbench. Initial accuracy evaluation done by the Deep Learning Workbench.

Simply choose the model, the target environment and the dataset on which to run the evaluation. And the workbench handles the rest. It will give you not only the accuracy, but also the FPS of the model for your chosen target environment. From there on, you decide whether to optimize it further or straightaway deploy.

4. Model Tuning and Quantization

Be it Default-Quantization or Accuracy-Aware-Quantization, the process of model quantization becomes easier and hassle-free with the DL workbench.

The screenshot shows the 'Quantization' tab in the Deep Learning Workbench. At the top, there is a dropdown for 'Subset Size, %' set to 10. Below it, the 'Optimization Methods' section has two radio buttons: 'Default Method' (selected) and 'AccuracyAware Method'. The 'Default Method' section notes 'Uncontrollable minor drop of model accuracy', 'Significant increase of model speed', and 'Annotated or not annotated datasets'. The 'AccuracyAware Method' section notes 'Controllable drop of model accuracy', 'Increase of model speed', and 'Annotated datasets only'. A 'Max Accuracy Drop' slider is set to 0.5. In the 'Calibration Schemes' section, 'Performance Preset' (selected) is noted as 'Uncompromising performance', while 'Mixed Preset' is noted as 'Tradeoff between accuracy and performance'. At the bottom, there are 'Optimize' and 'Cancel' buttons, and a link to 'Quantization methods'.

It need to be followed a series of manual steps to convert an FP32-Tiny YOLOv4 model into an INT8-precision model. Besides having to take care of each step and configuration file, we also had to ensure that each path and command was correct. With the DL workbench, you need not worry about any of these. Just provide the FP32 model, the dataset, and the desired accuracy. Within minutes, you will have an optimizer and a quantized model ready for use.

5. Integration and Deployment

Lastly, the DL workbench also provides a final-deployable model which you can just click and export. The exported, deployable package will contain:

- The optimized model
- All the configuration files
- The results of the various runs and experiments that you carry out

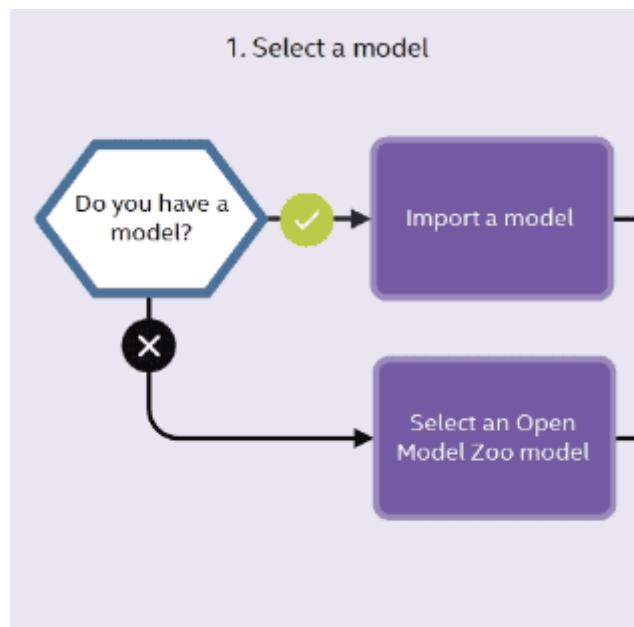
With all this information, you can easily zero in on the best possible way to deploy your model, and know exactly how it will perform. You can integrate it within an application, deploy it on the edge, or simply decide to run inference using the model.

Workflow of the Deep-Learning Workbench

The following stages presents the workflow of the Deep-Learning workbench, illustrating all the steps, starting from model selection right up to model deployment. The general workflow consists of 7 steps. Let's break these down into different components for greater clarity.

1. Model selection

You always start by selecting the model you want to optimize.

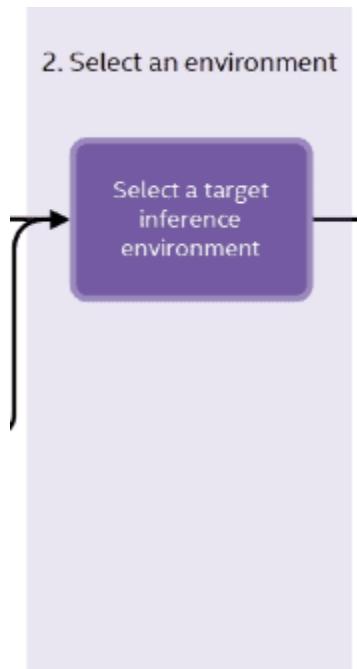


[Model Selection](#)

You are not constrained to choose a model from a specific framework, just ensure it comes from any of the OpenVINO-supported frameworks. You are even free to choose an OpenVINO-IR format model, provided you have already converted it using the Model Optimizer from any of the above frameworks.

2. Selecting the Target Environment

Next, select the target environment. This is important because the DL workbench will then optimize the model to run best on this particular hardware environment.



[Target Selection](#)

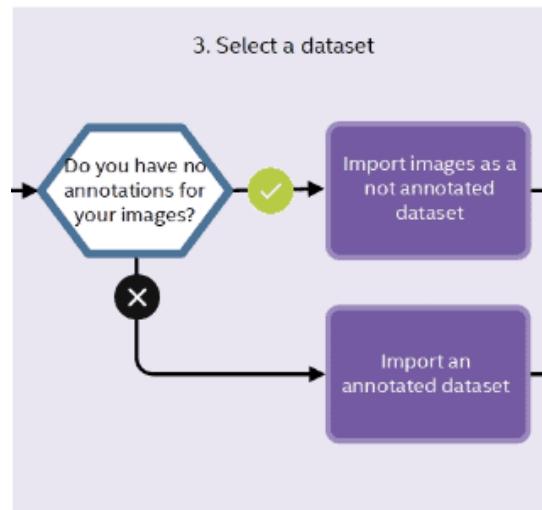
The target environment can be:

- a CPU
- the Intel Integrated GPU
- even VPUs like the Myriad X

In fact, you can even target a remote environment, which is not local to the system in which you are running the DL workbench.

3. Selecting the Dataset

Now that you've chosen the environment, select the dataset on which you want to run the evaluation and optimize the model.

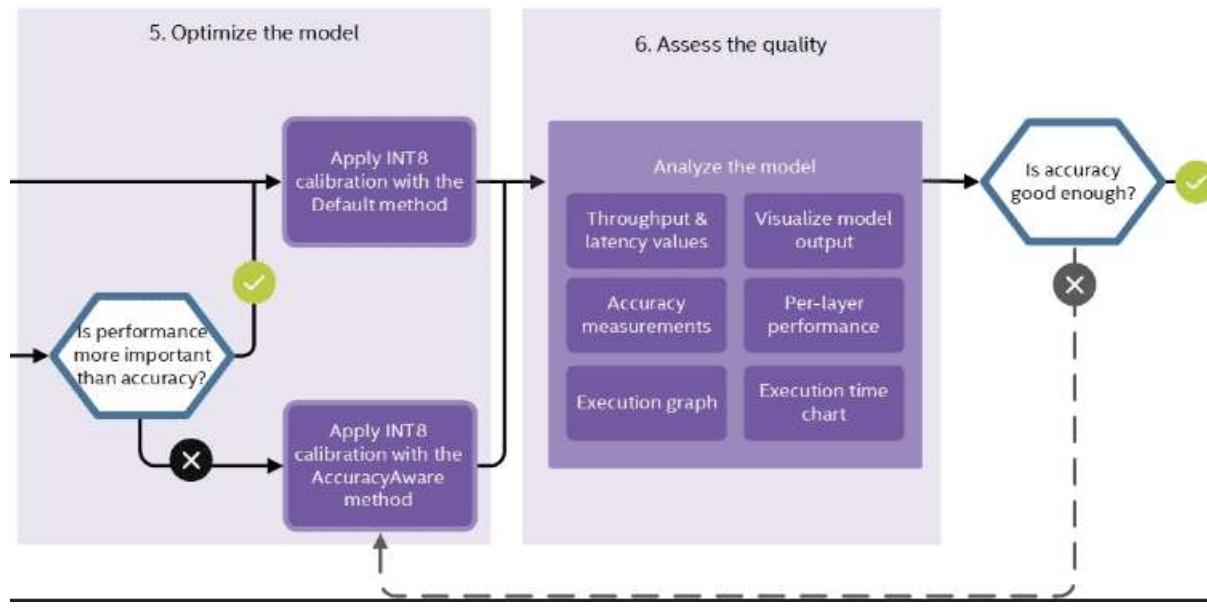
[Dataset Selection](#)

The DL workbench supports a number of datasets, including:

- Object-detection datasets like MS COCO, Pascal VOC
- Image-classification datasets like the ImageNet dataset
- Common Semantic Segmentation (CSS) dataset for semantic segmentation
- Common Super Resolution (CSR) dataset for super resolution, image inpainting and style transfer

4. Model Optimization

After model and dataset selection, optimize your model.

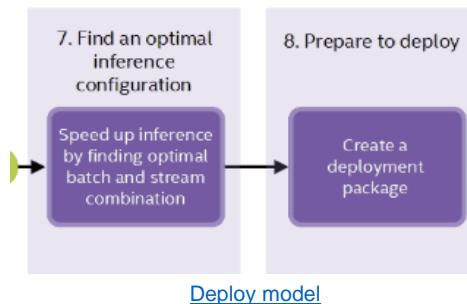


Apply quantization to convert the FP32 models into INT8-precision models. Also,

assess the precision of your model, so that you know for sure it will perform well in the real-world.

5. Configure and Deploy

The final few steps will configure the model according to your requirements and deploy it.



Experiment with different batch sizes and compare the throughput across different runs. When you get the desired tradeoff between speed and throughput:

- create the deployment package
- download it
- deploy it on an edge device

4.10 Utilizing OpenVINO for inference at the edge

Installing Deep-Learning Workbench Through Docker

Only after you install the Deep-Learning workbench in your system can you use it for optimization. The easiest way to install it is through the Docker Hub. First, install the Docker Engine, that being a necessary prerequisite. Please follow the instructions given here to install the Docker Engine on Ubuntu. Now, follow these steps to install the DL workbench through Docker **Hub on Linux (Ubuntu 20.04)**:

Step 1: Go inside the workbench Folder in the OpenVINO Installation Directory

```
1 | cd /opt/intel/openvino_2021.3.394/deployment_tools/tools/workbench
```

Step 2: Download the Workbench Starting Script

To start the DL workbench, you need to download the script. Give the following command to download the starting script for DL workbench, inside the current working directory.

1 | wget https://raw.githubusercontent.com/openvinotoolkit/workbench_aux/master/start_workbench.sh

Step 3: Ensure the File is Executable

In many cases, execution permissions are disabled by default for security reasons. Execute the following command to ensure the script is executable.

1 | chmod +x start_workbench.sh

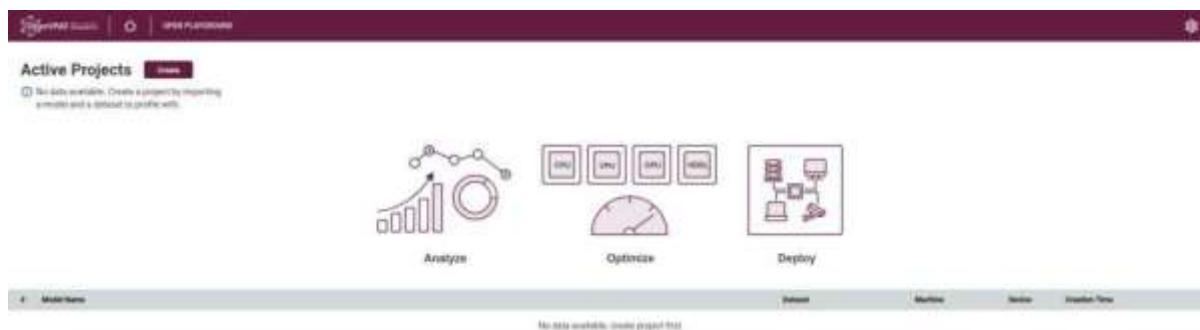
Step 4: Run the DL Workbench

Finally, start the DL workbench through the below command execution terminal. ./start_workbench.sh

You may need to wait for some time till the DL workbench is ready. After the script runs all the commands successfully, a prompt on the terminal informs you that the DL workbench is available on the local host at port 5665.

<http://127.0.0.1:5665/>

Open the link and you will see the following browser window:



The initial window of the DL Workbench.

With this, you have completed the installation of DL workbench.

Applying AccuracyAwareQuantization to Tiny YOLOv4, Using DL Workbench

Just follow these steps:

Step 1: Start the DL Workbench

First, open the DL workbench, by following all the steps discussed in the installation section.



You will see the **Create** button at the top of the initial DL workbench window. Click on it.

Step 2: Create Project

That takes you to the **Create Project** page, which should look similar to this:

[Start Page](#) / [Create Project](#)

Create Project

- ⓘ Select a model, dataset, and environment.
Then click Create to perform an inference.

Project Details

- ✗ Model: Selection Required
- ✗ Target: Selection Required
- ✗ Device: Selection Required
- ✗ Dataset: Selection Required

Model ^ [Import](#)

Model Name

- ⓘ To continue working, import a model.

Environment ^ [Add Remote Target](#)

ⓘ Target Platform

Target Name

Available Devices

Local Workstation

CPU

You need four things to successfully optimize a model, using the DL workbench:

- The Deep-Learning model
- The target environment
- The target device or hardware
- An evaluation dataset

Initially, all these options will be marked with red crosses (as in the above image), indicating that none of the requirements are met.

Step 3: Import the Deep-Learning Model

Next, import the Deep-Learning model. For this example, we will be importing the Tiny-YOLOv4 FP32 model, which is already in the IR format. So no need to run it through the model optimizer. Simply, click on the Import button, which should take you to the following screen:

Import Model

1. Import

[Open Model Zoo](#)
[Original Model](#)

Framework:

OpenVINO™ IR

IR XML file: [?](#)

Select

frozen_darknet_yolov4_model.xml

IR BIN file: [?](#)

Select

frozen_darknet_yolov4_model.bin

Model name: [?](#)

frozen_darknet_yolov4_mode

Import Model

Cancel

You will need both:

- The .xml file containing the network topology
- The .bin file containing the model weights

Finally, click on the **Import Model** button. The model might take some time to upload to the DL-workbench environment.

Step 4: Select the Target Environment and Hardware

Then you need to select the target environment and hardware.

Processor Details	Clock Speed	Available Cores	Platform Tag	Intelligence Details
All available models: Local CPU (CPU)	1.60 GHz (1.70 GHz)	4 cores (7.07 GHz)	CPU	4 cores - 0.0-1.000

Selecting the target environment and hardware.

For this example, we are using the **Local Environment** and the local CPU as our targets. When running on your local system, you should see your own CPU model and can choose accordingly.

Step 5: Import the Evaluation Dataset

The final step before carrying out evaluation and optimization is to choose the evaluation dataset. If you just needed to carry out the evaluation, you wouldn't have to import any official dataset. Just creating a random dataset in the DL workbench would do. But here we need to quantize the model as well, so we will require a validation subset on which the quantized model can be evaluated.

[Start Page](#) / [Create Project](#) / Import Dataset

Import Validation Dataset

- ⓘ Import an dataset in one of the supported formats: [ImageNet](#), [Pascal VOC](#), [COCO](#), [Common Semantic Segmentation](#), [Common Super-resolution](#), [LFW](#), [VGGFaces2](#), or [not annotated](#).

Select File: coco.zip

Imported Dataset Name: [?](#)

Here, we are importing the MS COCO 2017 validation dataset for evaluation purposes. This zip file contains 5000 validation images in total, along with their corresponding images. To download the dataset from the official website, click [here](#). After this, you should see a *green tick mark* across all the requirements.

Create Project

- ① Select a model, dataset, and environment.
Then click Create to perform an inference.

Project Details

- ✓ Model: frozen_darknet_yolov4_model
- ✓ Target: Local Workstation
- ✓ Device: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
- ✓ Dataset: coco

These are all the things you need to start the model evaluation/optimization. Next, click on the **Create** button on the browser window to start the process.

Initial Run Results

After this, run one initial evaluation on the entire validation dataset, using the full-precision model we selected above. Check out the results in the following image:

↗ Expand Table						
#	Stream	Batch	Throughput, FPS	Latency, ms	Last Status	Display Experiment
A	1	1	27.6	35.17	✓	<input checked="" type="checkbox"/>

The above results are from the i7 CPU machine, as mentioned in the previous section. Your results may vary depending on the hardware. For the FP32 model, the initial is giving 27.6 FPS on average and latency of 35.17 milliseconds. It will be interesting to compare these results with the quantized model, after applying AccuracyAwareQuantization.

Step 6: Optimize Performance

To start the AccuracyAwareQuantization:

- Click on the **Perform** tab on the current screen,
- Then click the **Optimize Performance** button. This should let you select the INT8-optimization method.



Selecting the optimization method for quantization. Next, click the **Optimize** button.

Step 7: Configure the Accuracy Settings and Select Dataset-Subset Size

Now you need to configure the accuracy settings. It is safe to leave the settings to default value initially. You should see a screen similar to this:



Configuring the accuracy settings for AccuracyAwareQuantization. For the optimized configuration settings:

- The metric is mAP (mean Average Precision)
- The dataset is COCO with 80 classes
- Both IOU (Intersection Over Union) and NMS (Non-Max Suppression) have 0.5 threshold
- The usage type is object detection, as we are using the Tiny-YOLOv4 model
- The *Model Type* is set to Tiny YOLOv2

As you must have noticed by now, instead of v3 or v4, the *Model Type* is Tiny YOLOv2. Because the recent Tiny YOLO versions are not-fully supported by OpenVINO, we are bound to choose the Tiny YOLOv2 as the Model Type. Though this will not cause issues while quantizing the model, it will fail to give us any accuracy results. So, we will be checking the accuracy manually, by using the COCO evaluator to run evaluation manually on the entire COCO-validation set. After completing the above settings, choose AccuracyAwareQuantization as the quantization method.

Subset Size, %: 20

Optimization Methods

Default Method
Uncontrollable minor drop of model accuracy
Significant increase of model speed
Annotated or not annotated datasets

AccuracyAware Method
Controllable drop of model accuracy
Increase of model speed
Annotated datasets only

Max Accuracy Drop: 0.5

Calibration Schemes

Performance Preset
Uncompromising performance

Mixed Preset
Tradeoff between accuracy and performance

Optimize Cancel

For this example, we have set the *Max Accuracy Drop* value to 0.5. This means, while doing INT8 calibration, if the accuracy drops below this specified threshold for any particular layer, then that layer will revert back to the original precision.

Also, take note of the dataset Subset Size. We are using just 20% of the 5000 images. So the calibration and evaluation will be done only on 1000 images. Selecting the whole dataset would have eaten up too much time, sometimes it takes hours to complete. Finally, you can click on the **Optimize** button.

Step 8: Check the Results

Let the calibration tool run the AccuracyAwareQuantization. It could take some time, depending on your hardware (CPU).

							↗ Expand Table
#	Stream	Batch	Throughput, FPS	Latency, ms	Last Status	Display Experiment	
A	1	1	60.48	17.66	✓	<input checked="" type="checkbox"/>	

We got:

- 60 FPS, in terms of throughput
- The latency dropped to 17.66 milliseconds, after the INT8 calibration

This is a huge improvement compared to the 27 FPS and 35 milliseconds latency seen in the case of the full precision model. We can also check the **Precision Distribution** in our calibrated model to ensure the model was actually converted to the INT8 format.

Precision Distribution

Precision	Execution Time, %	Include to Distribution Chart
I8	97.24	<input checked="" type="checkbox"/>
FP32	2.76	<input checked="" type="checkbox"/>
BOOL	0.00	<input checked="" type="checkbox"/>
FP16	0.00	<input checked="" type="checkbox"/>

You can see that:

- More than 97% of the execution time was spent in the INT8 layers
- Around 2.7% of the time was spent in the FP32 layers

Okay, so this means most of the layers have been successfully converted to INT8 precision. Only a few layers reverted back to their original FP32-precision format, probably because they exceeded the accuracy-drop threshold of 0.5.

Step 9: Export the Calibrated Model

The final step is to export your INT8-calibrated model so that you can run an inference with it.

Analyze Perform Details Open in Playground

Optimize Performance Explore Inference Configurations Visualize Output Create Deployment Package Export Project

Include Model
 Yes No

Include Dataset
 Yes No

Include Accuracy Configuration
 Yes No

Include Calibration Configuration
 Yes No

Go to the **Export Project** sub-tab on the Perform tab, choose the calibrated model, and click on the Export button. The downloaded file will contain the .xml as well as the .bin file.

Step 10: Run Inference Using INT8-Calibrated Tiny-YOLOv4 Model

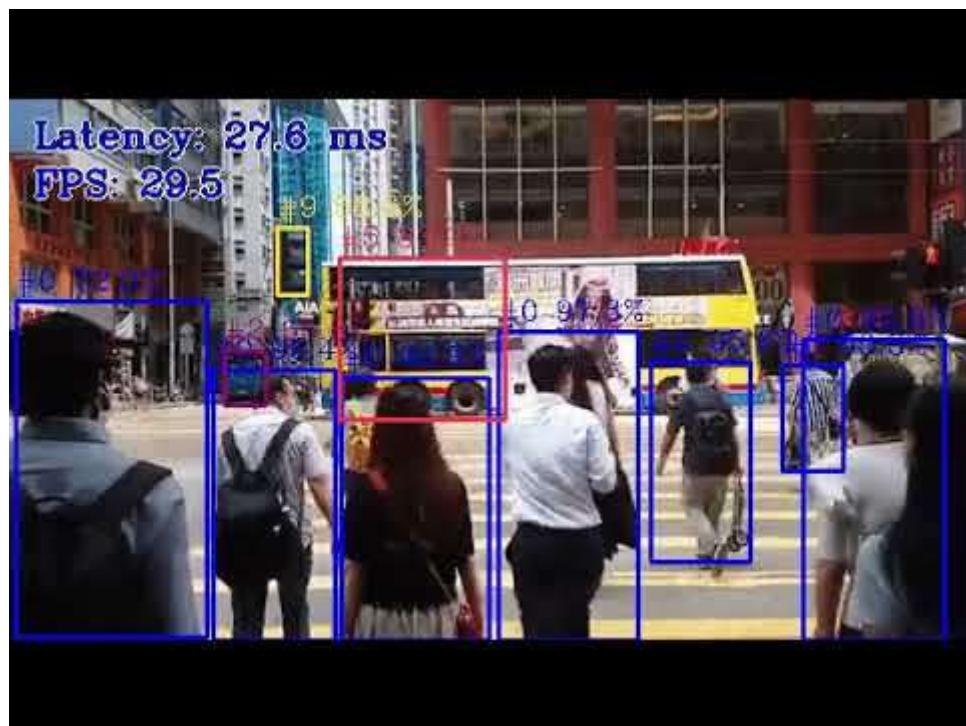
Now that we have obtained the INT8 models from the above steps, let's try running inference on the following video. To run inference, we use almost the same commands as in the previous posts in this series. The only difference being the path to the INT8-calibrated model. As the input video also remains the same, we can compare performance with the FP32 model.

```
1 | python object_detection_demo.py --model frozen_darknet_yolov4_model.xml  
| -at yolo -i video_1.mp4 -t 0.5 -o acc_aw_int8_default.mp4
```

For the above run:

- the average FPS is 30.3
- latency is 27.3 milliseconds

Now, check out the following video output:



The above results are really interesting. Not only are we getting throughput very similar to the Default Quantized INT8 model, but also the detections look exactly the same as the FP32 model. This means **we are getting good speed and accuracy at the same time**.

4.11 Edge Computing Using OpenVINO and Raspberry-Pi

Install OpenVINO on R-Pi

- Go to Downloads folder
- cd Downloads

- Install the OpenVINO (2020.4) toolkit using wget command

```
wget --no-check-certificate
https://storage.openvinotoolkit.org/repositories/openvino/packages/2020.4/I_openvino_toolkit_runtime_raspbian_p_2020.4.287.tgz
```

By default, the package file is saved as I_openvino_toolkit_runtime_raspbian_p_<version>.tgz.

Note: If the file isn't run by wget, use go to this link and download from GUI
Link : <https://storage.openvinotoolkit.org/repositories/openvino/packages/2020.4/>

- Create an installation folder.

```
sudo mkdir -p /opt/intel/openvino_2021
```

- Unpack the archive:

```
sudo tar -xf I_openvino_toolkit_runtime_raspbian_p_2020.4.287.tgz --strip 1 -C /opt/intel/openvino_2021
```

- Set the Environment Variables

You must update several environment variables before you can compile and run OpenVINO toolkit applications. Run the following script to temporarily set the environment variables:

```
source /opt/intel/openvino_2021/bin/setupvars.sh
```

***(Optional)** The OpenVINO environment variables are removed when you close the shell. As an option, you can permanently set the environment variables as follows:

```
echo "source /opt/intel/openvino_2021/bin/setupvars.sh" >> ~/.bashrc
```

- Add USB Rules for an Intel® Neural Compute Stick 2 device

This task applies only if you have an Intel® Neural Compute Stick 2 device.

Add the current Linux user to the users group:

```
sudo usermod -a -G users "$(whoami)"
```

To perform inference on the Intel® Neural Compute Stick 2, install the USB rules running the `install_NCS_udev_rules.sh` script:

```
sh /opt/intel/openvino_2021/install_dependencies/install_NCS_udev_rules.sh
```

- Log out and log in for it to take effect. (Reboot)

```
sudo reboot
```

To test your change, open a new terminal. You will see the following:

```
[setupvars.sh] OpenVINO environment initialized
```

- Plug in your Intel® Neural Compute Stick 2 in USB 2.0
- **Checking for Movidius Device (VPU)**

```
dmesg | grep Movidius
```

You will get the following output indicating that we have Movidius MyriadX connected to Raspberry Pi

```
$ dmesg | grep Movidius
[ 2.062235] usb 1-1.2: Product: Movidius MyriadX
[ 2.062244] usb 1-1.2: Manufacturer: Movidius Ltd.
$
```

Face Detection using OpenVINO on R-Pi

4.12 Practical: Face Detection

- Clone the github repository into local system

```
git clone https://github.com/ameer-aiml/face-detect-ov-rpi
```

This repository contains both xml and bin file of Face Detection model.

- Go to Face Detection folder

```
cd face-detect-ov-rpi /Face Detection
```

Note: Add a single before and after Face Detection wording if not recognized

- Run the python file (it should be python3)

python3 face_detection.py

Method 2: If git clone isn't working

- Create a folder named face
`mkdir face && cd face`
- Download xml and bin file using wget for Face Detection Retail from web browser and move it to face folder

Link : https://download.01.org/opencv/2019/open_model_zoo/R1/models_bin/face-detection-retail-0004/FP16/

NOTE: Take only FP16 because VPU works only on FP16 .

- Copy and paste the face_detection.py file in face folder of Raspberry Pi
- Run python file using python3 command
`python3 face_detection.py`

Errors which might be faced:

- `cv2.error: OpenCV(4.4.0-opencvino)/opencv/modules/dnn/src/ie_ngraph.cpp:638: error: (-2:Unspecified error) Failed to initialize Inference Engine backend (device = MYRIAD): Can not init Myriad device: NC_ERROR in function 'initPlugin'`

Either VPU isn't working or you are using VPU in USB 3.0 (Connect it to USB 2.0)

4.13 Practical: Face, Age & Gender Detection using OpenVINO & R-PI

- Clone the github repository into local system

git clone <https://github.com/ameer-aiml/age-gender-openvino-rpi>

- Go to Age Gender Detection folder

cd age-gender-openvino-rpi/AgeGender

This repository contains 3 different models.

One for Face Detection, another for Age and last one for Gender Detection.

Both the files of Face Detection is available. The .caffemodel files of Age and Gender are missing.

- Download the Age and Gender Caffe model from this site using the command
wget
(Github doesn't support more than 25 mb file)

wget

https://www.dropbox.com/s/iyy483wz7ztr9gh/gender_net.caffemodel

wget https://www.dropbox.com/s/xfb20y596869vbb/age_net.caffemodel

- Run the python file (it should be python3)

python3 face_detection.py

Errors which might be faced:

- **cv2.error: OpenCV(4.4.0-opencvino)
..../opencv/modules/dnn/src/ie_ngraph.cpp:638: error: (-2:Unspecified
error) Failed to initialize Inference Engine backend (device = MYRIAD):
Can not init Myriad device: NC_ERROR in function 'initPlugin'**

Either VPU isn't working or you are using VPU in USB 3.0 (Connect it to USB 2.0)

Challenge to try out:

Convert all the above mentioned 6 files into xml and bin file respectively using model optimizer for unified way of model building.

If you want to use your model for inference, the model must be converted to the .bin and .xml Intermediate Representation (IR) files that are used as input by Inference Engine. OpenVINO™ toolkit support on Raspberry Pi only includes the Inference Engine module of the Intel® Distribution of OpenVINO™ toolkit. The Model Optimizer is not supported on this platform.

Module IV

SAP Business

Technology

Platform ABAP

Environment

Unit 1: Introduction to SAP Ecosystem

Learning Outcomes:

- Basic Understanding of SAP ERP
- Evolution of SAP ERP
- Understanding of SAP Layered Architecture

1.1 What is SAP

SAP is one of the world's leading producers of software for the management of business processes, developing solutions that facilitate effective data processing and information flow across organizations.

Overview

Founded in 1972, the company was initially called System Analysis Program Development. Since then, it has grown from a small, five-person endeavor to a multinational enterprise headquartered in Walldorf, Germany, with more than 105,000 employees worldwide.

SAP established the global standard for enterprise resource planning (ERP) software. Now, SAP S/4HANA takes ERP to the next level by using the power of in-memory computing to process vast amounts of data, and to support advanced technologies such as artificial intelligence (AI) and machine learning.

The company's integrated applications connect all parts of a business into an intelligent suite on a fully digital platform, thereby replacing the process-driven, legacy platform. Today, SAP has more than 230 million cloud users, more than 100 solutions covering all business functions, and the largest cloud portfolio of any provider.

Understanding SAP

What does SAP stand for?

The name is an initialism of the company's original German name: Systemanalyse Programmentwicklung, which translates to System Analysis Program Development. Today the company's legal corporate name is SAP SE — SE stands for societas

Europaea, a public company registered in accordance with the European Union corporate law.

What is SAP software used for?

Traditional business models often decentralise data management, with each business function storing its own operational data in a separate database. This makes it difficult for employees from different business functions to access each other's information. Furthermore, duplication of data across multiple departments increases IT storage costs and the risk of data errors.

By centralising data management, SAP software provides multiple business functions with a single view of the truth. This helps companies better manage complex business processes by giving employees of different department's easy access to real-time insights across the enterprise. As a result, businesses can accelerate workflows, improve operational efficiency, and raise productivity.

What does SAP do?

SAP helps companies and organizations of all sizes and industries run their businesses profitably, adapt continuously, and grow sustainably.

The company develops software solutions that are used by small businesses, midsize companies, and large corporations. With standard applications, industry solutions, platforms, and technologies, every business process can be mapped and designed. The software collects and processes data on one platform, from raw material purchasing to production and customer satisfaction.

In addition, SAP helps customers seamlessly link operational data on business processes with experience data on emotional factors such as purchase experience and customer feedback. This enables companies to better understand and respond to their customers.

1.2 ERP Systems

What is ERP Software?

ERP stands for “enterprise resource planning.” ERP software includes programs for all core business areas, such as procurement, production, materials management, sales, marketing, finance, and human resources (HR).

SAP was one of the first companies to develop standard software for business solutions and continues to offer industry-leading ERP solutions.

Enterprise resource planning systems are complete, integrated platforms, either on-premises or in the cloud, managing all aspects of a production-based or distribution business. Furthermore, ERP systems support all aspects of financial management,

human resources, supply chain management, and manufacturing with your core accounting function.

ERP systems will also provide transparency into your complete business process by tracking all aspects of production, logistics, and financials. These integrated systems act as a business's central hub for end-to-end workflow and data, allowing a variety of departments to access.

How does it work?

ERP has evolved over the years from traditional software models that made use of physical client servers and manual entry systems to cloud-based software with remote, web-based access.

Businesses select the applications they want to use. Then, the hosting company loads the applications onto the server the client is renting, and both parties begin working to integrate the client's processes and data into the platform.

Once all departments are tied into the system, all data is collected on the server and becomes instantly available to those with permission to use it. Reports can be generated with metrics, graphs, or other visuals and aids a client might need to determine how the business and its departments are performing.

Benefits of Enterprise Resource Planning

Businesses employ enterprise resource planning (ERP) for various reasons, such as expanding, reducing costs, and improving operations. The benefits are :-

1. Improves Accuracy & Productivity

Integrating and automating business processes eliminates redundancies and improves accuracy and productivity.

2. Improves Reporting

Some businesses benefit from enhanced real-time data reporting from a single source system. Accurate and complete reporting help companies adequately plan, budget, forecast, and communicate the state of operations to the organization and interested parties, such as shareholders.

3. Increases Efficiency

ERPs allow businesses to quickly access needed information for clients, vendors, and business partners. This contributes to improved customer and employee satisfaction, quicker response rates, and increased accuracy rates.

4. Increases Collaboration

Departments are better able to collaborate and share knowledge; a newly synergized workforce can improve productivity and employee satisfaction as employees are better able to see how each functional group contributes to the mission and vision of the company.

Popular ERP Systems

1. SAP Business One

Increase control over your small business with software designed to grow with you. Streamline key processes, gain greater insight into your business, and make decisions based on real-time information – so you can drive profitable growth. Lower the cost of managing your business, from financials, purchasing, inventory, sales, and customer relationships to project management, operations, and HR.

2. SAP S/4 HANA Cloud

SAP S/4HANA Cloud is a complete, modular cloud ERP software designed for every business need – powered by AI and analytics. With SAP S/4HANA Cloud, you can run your mission-critical operations in real time from anywhere, introduce new business models in your industry, and expand globally with a trusted partner. For 50 years, SAP has been proudly helping enterprises of all sizes – in all industries and geographies – run at their best with ERP.

3. SAP ERP

Leverage world-class ERP software. Sharpen your competitive edge and drive growth with enterprise resource planning from SAP. With more than 40 years of experience and nearly 50,000 customers, our market-leading enterprise resource planning (ERP) software is a proven, trusted foundation built to support the world's largest organizations as well as small and midsize companies in 25 different industries.

1.3 Evolution of SAP

To handle the requirements of complex industries, SAP ERP solutions consolidate various Business processes and Operations. Most common Business Process and Operations are Sales & Distribution, Materials Management, Production Planning, Logistics Execution, and Quality Management), Financials (Financial Accounting, Management Accounting, Financial Supply Chain Management), Human Capital Management (Training, Payroll, e-Recruiting) and Corporate Services (Travel Management, Environment, Health and Safety, and Real-Estate Management).

The journey of SAP ERP from R/1 system to SAP S/4 HANA system

1. SAP R/1 1972

SAP entered the ERP software domain with its SAP R/1 system. R1 stands for single tier architecture. All layers Presentation, Application and Database are placed in the same system.

2. SAP R/2 1982

Based on mainframe architecture, SAP R/2 system was designed to manage large global enterprises. R/2 stands for 2 tier architecture.

3. SAP R/3 and SAP ECC: 1992

R/3 followed client-server architecture. R/3 systems could also take advantage of then evolving internet technology. R/3 stands for 3-tier architecture with well-defined separate Database layer, Application layer and Presentation layer. The three layers are connected to each other with networks.

ECC stands for Enterprise Central Component and is successor of SAP R/3 system. With ECC the popularity of SAP as a business software provider increased manifold. Underneath SAP NetWeaver technology made ECC robust and scalable.

4. SAP S/4HANA: 2015

Launched in the year 2015, SAP S/4HANA is SAP's next generation business suite designed to work in a truly connected digital world.

"S" in 'S/4HANA' stands for Simple and 4 represents 4th generation; the complete name is SAP Business Suite 4 for SAP HANA (S/4HANA).

Evolution of SAP

1972

- Founded in the year 1972 by five ex-IBM employees Dietmar Hopp, Hasso Plattner, Hans-Werner Hector, Klaus Tschira, and Claus Wellenreuther, SAP was headquartered in Weinheim, Germany.
- At present, SAP is a multinational software corporation headquartered in Walldorf, Germany.
- SAP is the abbreviation of Systems, Applications, and Products in Data Processing.

1972- Developing Mainframe programs

- The five engineers were working on developing mainframe programs for payroll and accounting.
- SAP's first customer was Imperial Chemical Industries in Östringen.
- 1973, SAP releases its first product—a financial accounting system called RF. R over here stands for real-time.

1973 to 1979

- Other systems were under development, and together they were called SAP R/1.
- SAP achieved purchasing, inventory management, and invoice verification with SAP's RM system
- During this phase, SAP worked on IBM servers and DOS operating system.

1979, SAP launched SAP R/2

- SAP R/2 was a mainframe software application capable of integrating all of an enterprise's business functions, including material management and production planning, with real-time processing.
- SAP now had 200 customers both inside as well as outside of Germany.

1992, SAP releases the new SAP R/3

- SAP R/3 was built on the client-server concept, having a uniform graphical interface, dedicated use of relational databases, and support for servers from various manufacturers.
- SAP sets on to tap a bigger market, including large as well as midsize companies.

1993, begins work with Microsoft

- 1993 SAP begins working with Microsoft, the world's largest software maker, to port SAP R/3 to the Windows NT operating system.
- 1994 The SAP R/3 system is released for Windows NT.
- 1996, SAP goes online with its Internet strategy with Microsoft. Through open interfaces, customers can now connect online applications to their SAP R/3 systems.

1999, mySAP.com

- In May 1999, SAP announces a new strategy that completely realigns the company and its product portfolio: mySAP.com.
- mySAP.com combines e-commerce solutions with SAP's existing ERP applications on the basis of cutting-edge Web technology.

2004- SAP NetWeaver

- SAP becomes the third-largest independent software provider in the world and a paragon of the German economy. The SAP brand stands for high-quality business software. SAP has now more than 24000 customers spread in over 120 countries.
- R/3 was replaced with the introduction of SAP ERP Central Component (ECC) 5.0 in 2004.
- 2004, SAP launches SAP NetWeaver to market. SAP NetWeaver brings in a new integration and application platform and a service-oriented architecture on top of which several SAP applications were build SAP ERP, SAP CRM, SAP SRM, etc.

2009, SAP Business Suite 7

- 2009, SAP introduces a new business suite, SAP Business Suite 7 software, designed to optimize business performance and reduce IT costs.

2011 on wards

- SAP continues to innovate and invest in cloud computing, an in-memory database called as SAP HANA database
- 2015, SAP launches its 4th Generation business suite, SAP S/4HANA, and SAP C/4HANA. This intelligent suite helps customers become Intelligent Enterprises.
- SAP strives towards becoming a leader in cloud computing and e-commerce business networks.

Focus on cloud – 2012

- To have a plethora of cloud-based products, since 2012, SAP has acquired several companies that sell cloud-based products. Acquisition of Concur Technologies, a provider of cloud-based
- Travel and expense management software in 2014 has been SAP's one of the costliest acquisitions. Other remarkable acquisitions are Success Factors, Ariba, Hybris, and FieldGlass. All these companies are providers of cloud-based products.
- In 2014, IBM and SAP began a partnership to sell cloud-based services.
- Likewise, in 2015, SAP also partnered with HPE to provide secure hybrid cloud-based services running the SAP platform. Both HPE and IBM provide infrastructure services to SAP, and SAP runs its SAP HANA cloud solution on top.

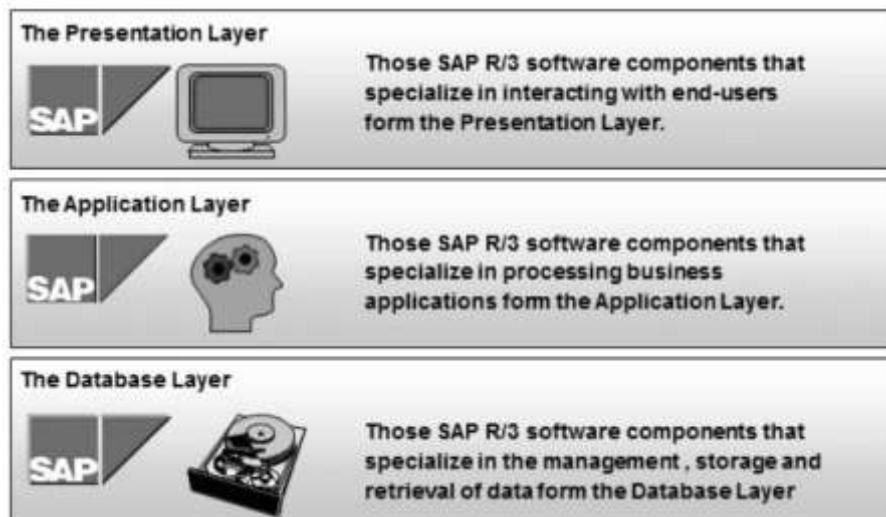
SAP S/4HANA- 2015

- In 2015, SAP launched its 4th generation business suite SAP S/4HANA to leverage the features of the SAP HANA platform. Real-time analytics, on the fly-computation, completely re-defined data models supported by modern SAP Fiori UX are some of the features of SAP S/4HANA.

1.4 SAP Layered Architecture

SAP Three-Tier Architecture

With SAP R/3, SAP ushers in a new generation of enterprise software — from mainframe computing (client-server architecture) to the three-tier architecture of database, application, and user interface.

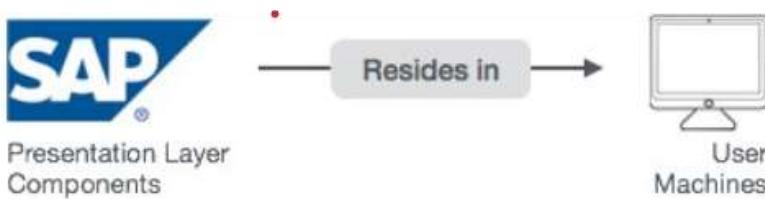


Reference: https://www.tutorialspoint.com/sap/sap_architecture.htm

Presentation Servers

Presentation servers contain systems capable of providing a graphical interface.

- Presentation Layer is also known as client Layer
- Presentation Layer is a user interaction
- In SAP-User interaction purpose we use GUI
- GUI stands for Graphical user interface
- Example – Desktop, Mobile Devices, laptops

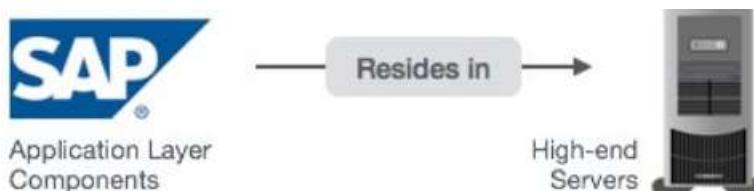


Reference: https://www.tutorialspoint.com/sap/sap_architecture.htm

Application Servers

Application servers include specialized systems with multiple CPUs and a vast amount of RAM.

- Application Layer is also known as Kernel Layer and Basic Layer.
- SAP application programs are executed in Application Layer.
- Application Layer serves as a purpose of a communicator between Presentation and Database Layer.
- Application server is where the dispatcher distributes the work load to the different work processes makes the job done.

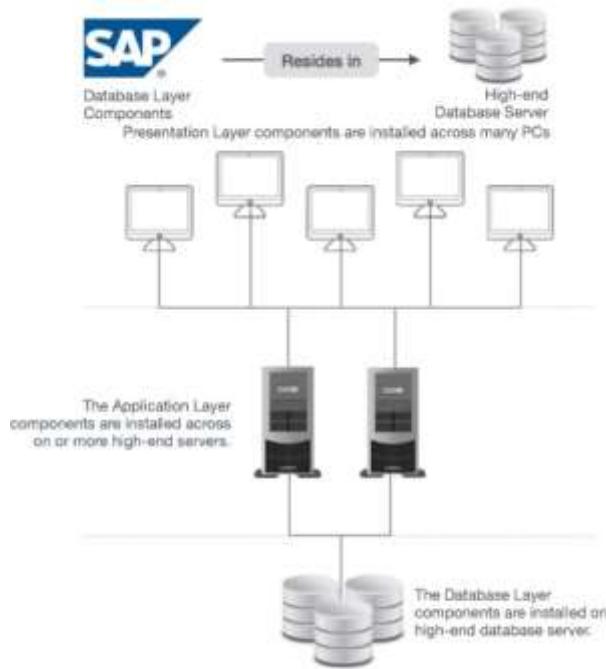


Reference: https://www.tutorialspoint.com/sap/sap_architecture.htm

Database Servers

Database servers contain specialized systems with fast and large hard-drives.

- Database layer stores the data
- Data store can be Business data, SAP system data, SAP tables, Programs.
- Examples – Oracle, Microsoft SQL Server, IBM DB/2, Siebel, Sybase, etc.



Reference: https://www.tutorialspoint.com/sap/sap_architecture.htm

1.5 Difference between SAP Functional & Technical Modules

SAP Functional Modules

In order to replicate and enable business process, SAP offers various predefined or standard functionality to help departments in performing various business activities. The SAP modules which provide predefined standard functionality to replicate actual business activity are called functional module. They process orders, track inventory, handle human resources, turn raw data to business intelligence and so on. A functional consultant has to understand the business requirement and use the standard functionality provided by SAP.

SAP Technical Modules

These modules do not directly replicate actual business activity but provide needed support to functional modules. They enable technical consultants to maintain and tune your landscape, schedule tasks, troubleshoot performance issues, build applications, download and install updates, and plan and execute migrations.

Unit 2: Introduction to SAP Functional ERP Module

Learning Outcomes:

- Basic Understanding of SAP BW, PP, MM, QS and PS Module
- Understanding different features in SAP functional ERP modules
- Concept of Screen Navigation

2.1 SAP PP (Production Planning)

2.1.1 What is Production Planning?

- Production Planning is the process of aligning demand with manufacturing capacity to create production and procurement schedules for finished products and component materials.
- SAP PP is an important module of SAP. It tracks and makes a record of the manufacturing process flows, for example, the planned and actual costs. Also, goods movements from the conversion of raw material to semi-finished goods.
- It is fully integrated with the other SAP modules: SD, MM, QM, FICO & PM.

2.1.2 Organization Structure in SAP PP

In any live Production Planning module, locations of manufacturing plants and storage within the plants, should be available in the system.

Importance of Plant and storage locations in Production Planning-

- All Production master data is created at Plant level.
- Planning activities are also performed at Plant level.
- Production Confirmation process and related goods movement occur at plant and storage location level.

2.1.3 Master Data in SAP PP

Master data is generally static for any company and is very rarely changed depending on the requirement. There are **5 master data** to be maintained in Production Planning module.



1. Material Master

The material master contain information on all the materials that a company procures, produces, stores, and sells. It is a number uniquely identifies a material master record, and hence a material.

Materials with the same basic attributes are grouped together and assigned to a material type such as finished, raw material, etc.

It is used for the following purposes:

1. To purchase materials
2. For Goods Movement postings such as goods issue or receipt in inventory management and also for physical inventory postings
3. In invoice verification for posting invoices
4. In sales and distribution for sales order fulfillment process
5. In production planning and control for material requirements planning, scheduling, and production confirmation processes.

2. Bill of Material (BOM)

A bill of material is a complete, formally structured list of the components together with the quantity required to produce the product or assembly.

BOM's are used in material requirement planning and product costing.

You can also create up to 99 alternative BOMs for a single product.

For Products having variants, you can create Super BOM, which has all possible types of components used to manufacture different types of variants, and the appropriate component is selected based on characteristic chosen in the sales Order.

For example, Product Cycle can contain all types of frames (with different colours and sizes) and desired frame is selected in production order based on colour and size chosen in the sales order.

3. Work Centre

A Work Centre is a machine or group of machines where production operations are performed. Work centres are used in task list operations (Routings).

It contains the data for

- Scheduling
- Capacity
- Costing

4. Routing

Routing is nothing but a sequence of operation performed at the Work Centre. It also specifies the machine time, labour time, etc. for the execution of operations.

It is also used for scheduling of operations and used in standard cost calculation of the product.

5. Production version

The production version is a combination of BOM and Routing data for production. It is a linkage between BOM & Routing and determines the manufacturing process. There can be multiple production versions as per different manufacturing process to produce the product

2.1.4 SAP PP – Common Tables

In this chapter, we will discuss some of the important tables in SAP PP.

For Material Requirement Planning Table	Description
MDKP	Document Header data
MDTB	Table Structure
MDVM	Planning File Details
MDFD	MRP Date details
S094	Stock Analysis

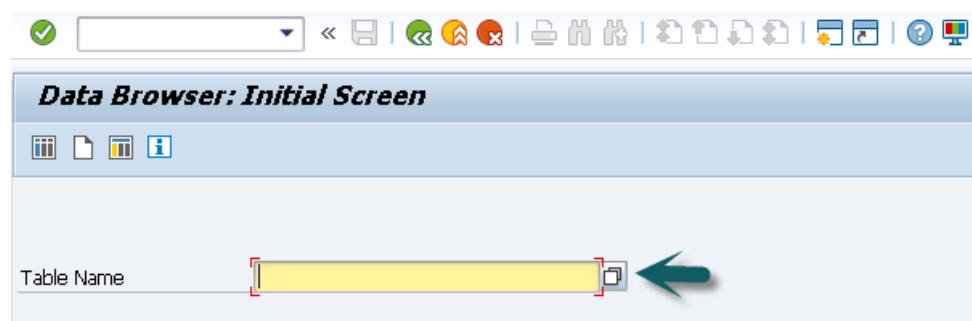
For Demand Management Table	Description
PBED	Independent Requirements Data
PBIM	Independent Requirements by Material

There are various tables in SAP PP system for BOM, routing, discrete production, material allocation, goods receipts, etc.

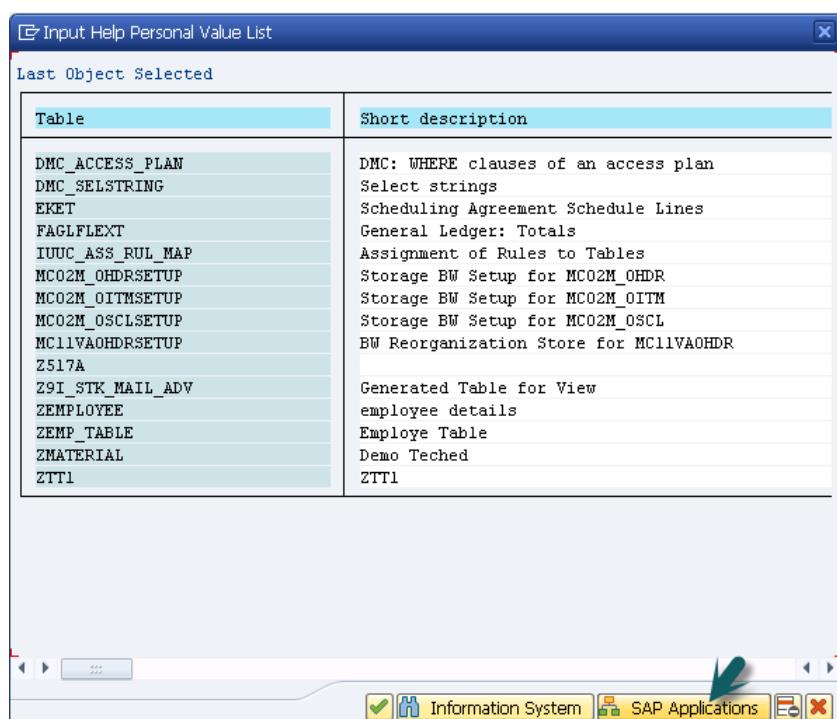
Table	Description
MAST	Material BOM
STKO	BOM Header
STOP	BOM Positions
PLKO	Routing Group Header
PLSO	Routing Group Sequence
PLPO	Routing Group Operations
AFKO	Production Order Header
AFPO	Production Order Position

How to View SAP PP Tables in SAP ERP System?

Step 1: In ERP system, use T-Code: SE16.



Step 2: Go to SAP Applications.



Step 3: Click the ‘+’ sign and you can see the list of all tables in ERP as shown in the following screenshot.

Table Name	Description
LE	Logistics Execution
QM	Quality Management
PM	Plant Maintenance
CS	Customer Service
PP	Production Planning and Control
ODATA_PP_SFC_PRODORDER_RELEASE	Backend Settings for Service prodorder.re
XLPO	Manufacturing: xLPO Integration
PP-BD	Basic Data
PP-SOP	Sales and Operations Planning
PP-MP	Production Planning
PP-CRP	Capacity Requirements Planning
PP-MRP	Material Requirements Planning
PP-MES	Integration with Manufacturing Execution
PP-SFC	Production Orders
PP-KAB	KANBAN
PP-REM	Repetitive Manufacturing
PP-PI	Production Planning for Process Industrie
PP-PDC	Plant Data Collection
PP-FLW	Flow Manufacturing
PP-IS	Information System
PP-PN	Production Network
PS	Project System
SCM	Supply Chain Management
EHS	Environment, Health and Safety

2.2 SAP MM (Material Management)

SAP MM is the short form for SAP Material Management system. The roles of SAP MM in a business process are as follows –

- A business process in SAP is termed as a “module”.
- SAP MM is a part of logistics functions and it helps in managing the procurement activities of an organization.
- It supports all aspects of material management (planning, control, etc.).
- It is the backbone of logistics that incorporates modules such as Sales and Distribution, Production Planning, Plant Maintenance, Project Systems, and Warehouse Management.

2.2.1 Features of SAP MM

The features of a SAP MM system are as follows –

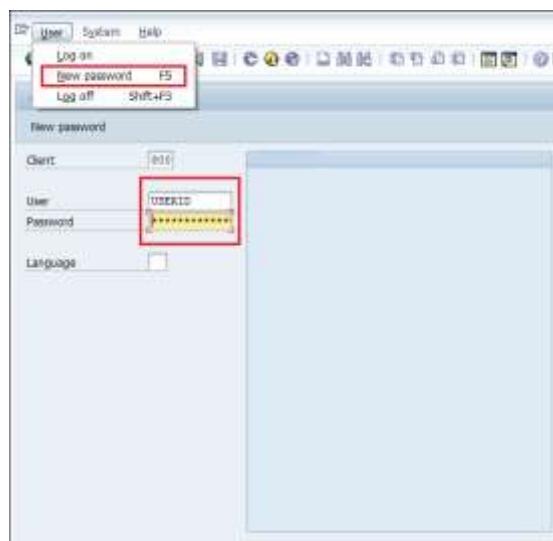
- SAP MM is one of the modules of SAP that deals with material management and inventory management.
- Material Management as a process ensures no shortage of materials or any gaps in the supply chain process of the organization. SAP MM speeds up the procurement and material management activities, making the business run smoothly with complete time and cost efficiency.

- It deals with managing the materials (products and/or services) and resources of an organization with the aim of accelerating productivity and reducing costs. At the same time, SAP MM is quite versatile to accommodate changes that are frequent in any business environment.
- It deals with the Procurement Process, Master Data (Material & Vendor Master), Account Determination & Valuation of Material, Inventory Management, Invoice Verification, Material Requirement Planning, etc.

2.2.2 Screen Navigation

Login Screen

Log on to the SAP ERP server. The SAP login screen will prompt you for the User ID and the Password. Provide a valid user ID and password and press enter. The user id and password are provided by the system administrator.

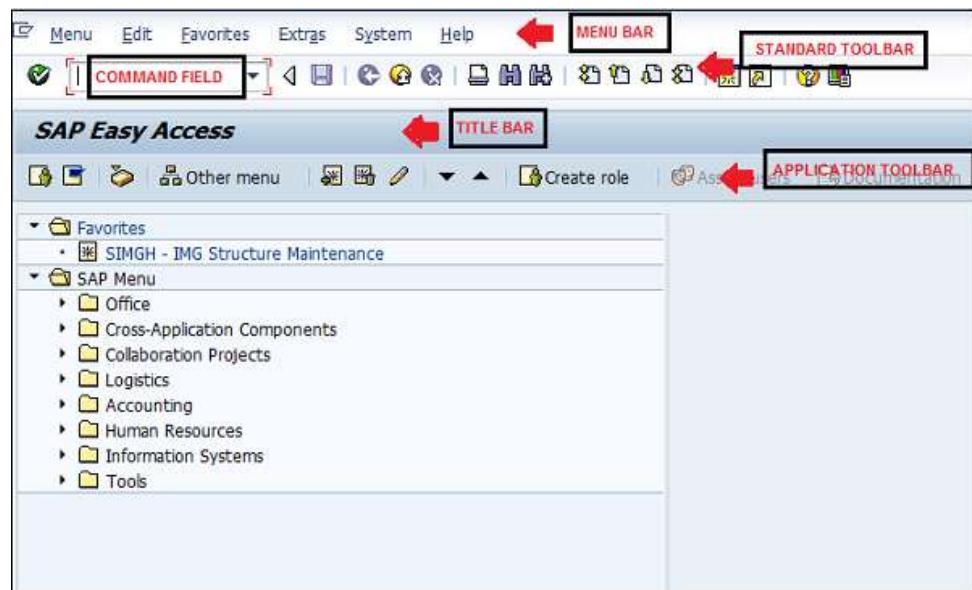


Reference: https://www.tutorialspoint.com/sap_mm/sap_mm_screen_navigation.htm

Standard Toolbar Icon

Given below is a brief description of the available toolbars –

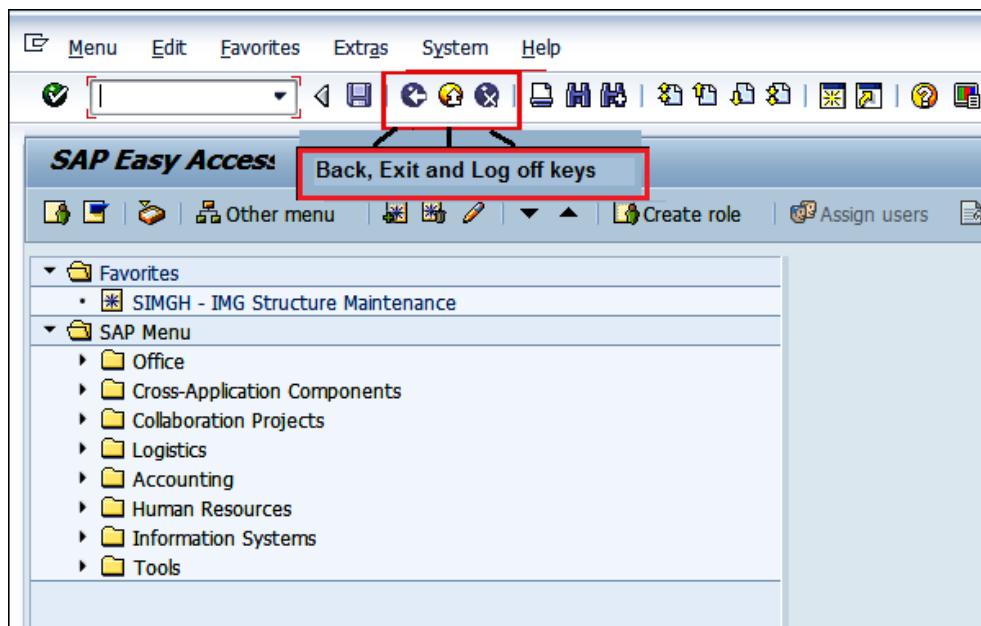
- **Menu Bar** – Menu bar is the topline of the dialog window in the SAP system.
- **Standard Toolbar** – This toolbar includes standard functions such as save, top of page, end of page, page up, page down, print, etc.
- **Title Bar** – Title bar displays the name of the application/business process you are currently in.
- **Application Toolbar** – Application-specific menu options are available on this toolbar.
- **Command Field** – To start a business application without navigating through menu transactions, some logical codes are assigned to the business processes. Transaction codes are entered in the command field to start an application directly.



Reference: https://www.tutorialspoint.com/sap_mm/sap_mm_screen_navigation.htm

Standard Exit Keys

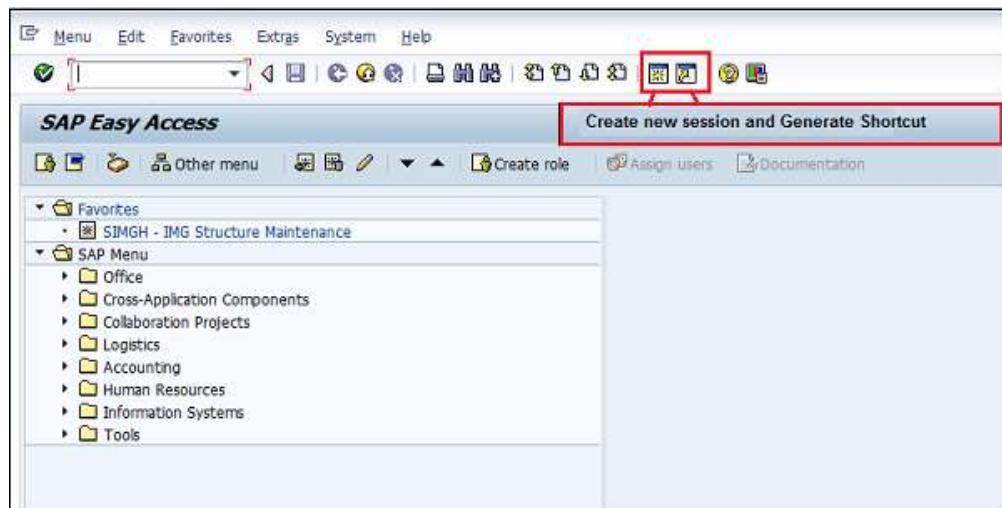
Exit keys are used to exit the module or to log off. They are used to go back to the last accessed screen.



Reference: https://www.tutorialspoint.com/sap_mm/sap_mm_screen_navigation.htm

New Session Icon

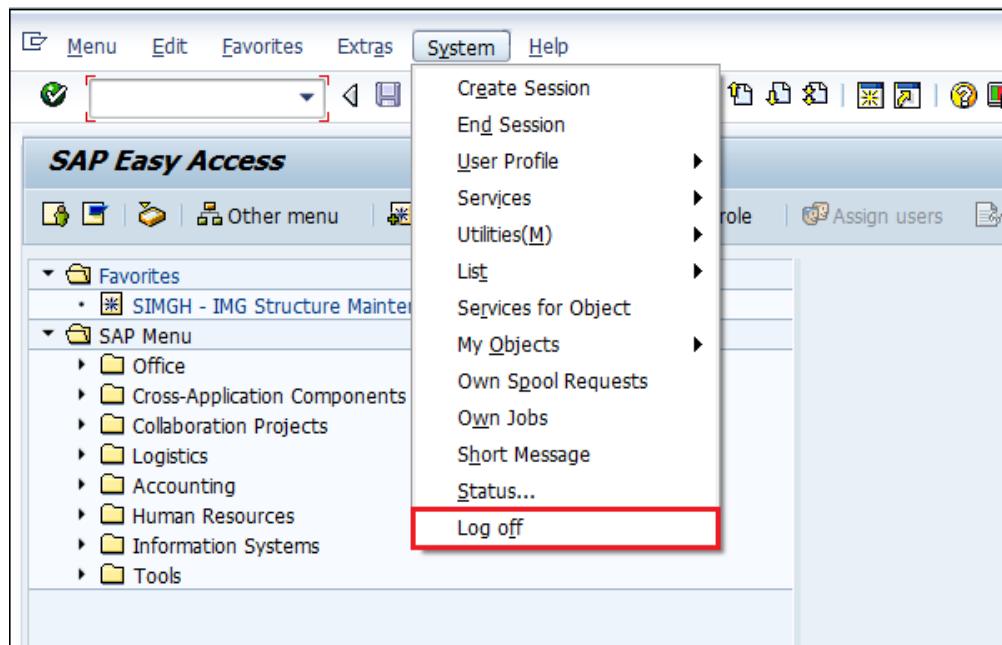
For creating a new session, we use the following key.



Reference: https://www.tutorialspoint.com/sap_mm/sap_mm_screen_navigation.htm

Log Off

It is a good practice to log off from the SAP system when you finish your work. There are several ways to log off from the system



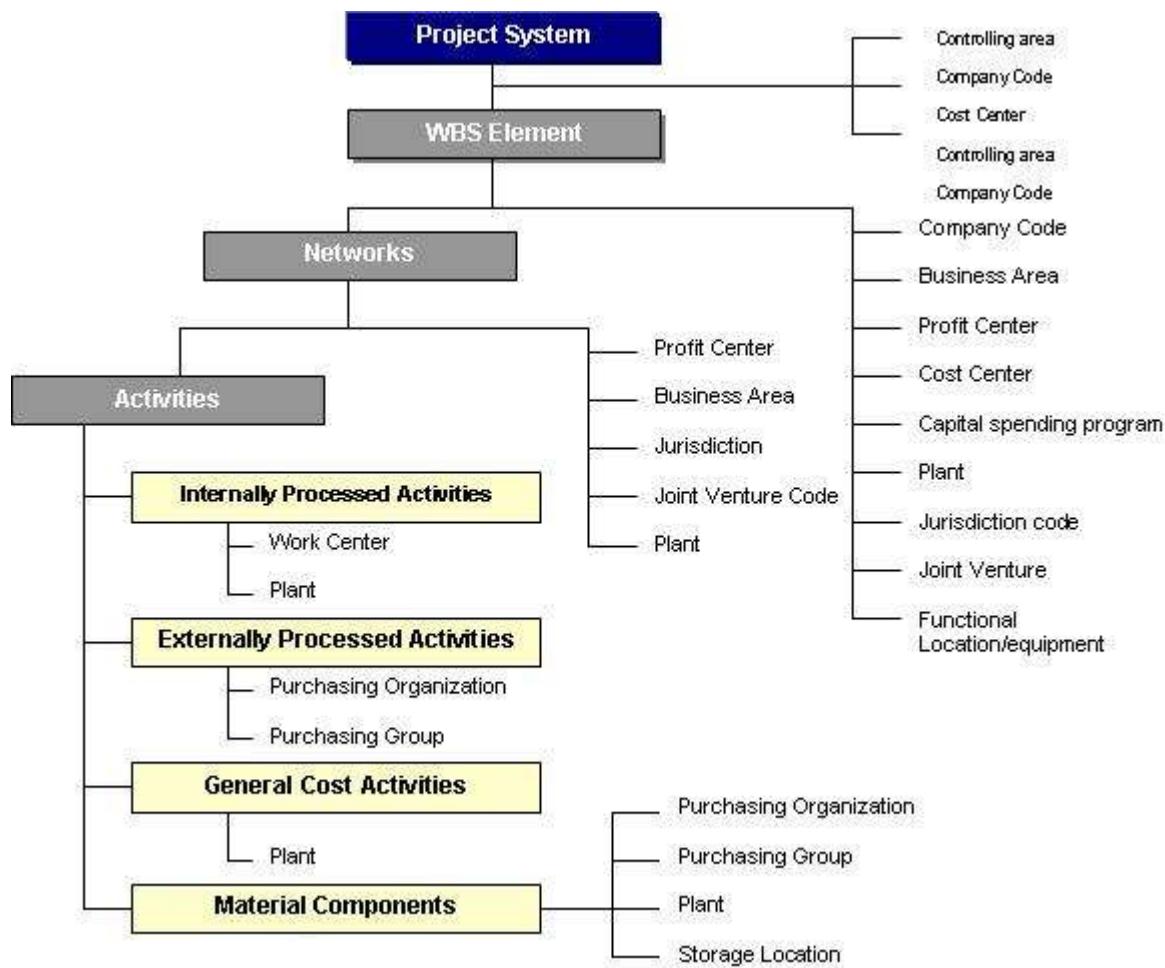
Reference: https://www.tutorialspoint.com/sap_mm/sap_mm_screen_navigation.htm

2.3 SAP PS (Project System)

The Project System (PS) module of SAP is specifically designed to provide comprehensive and fully integrated project management functionality for SAP customers. A Project is a complex undertaking, bringing with it huge data of different type.

At the beginning of each project, whether it involves developing a new product, make-to-order engineering, or internal organization, you need to define and set up the structures necessary to manage your project and incorporate them into your existing organizations and processes. Before you can run a project in its entirety, you must first describe the project goals precisely and create a structure for the project activities to be carried out. A clear project structure provides a basis for successful project planning, monitoring and control.

Organizational Structures in Project System:



You create and manage your project structure in SAP R/3 Project System, by means of work breakdown structures and networks. The WBS describes individual phases and functions of a project. Network contains the individual project tasks and the dependencies between them in the form of activities and relationships

Some important terms:

Project Definition

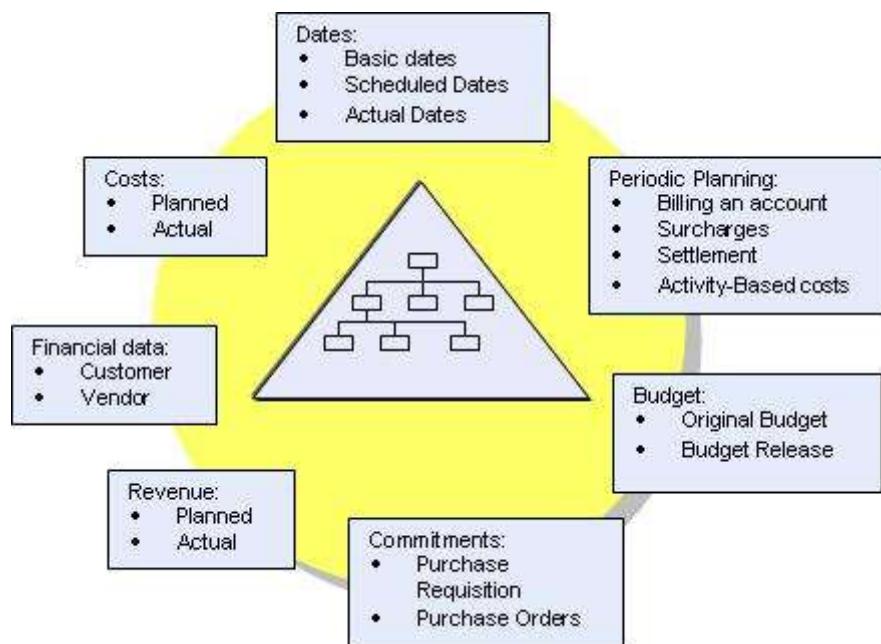
Project Definition is general description of the project that you want to manage. It is a framework laid down for all the objects created within a project. It contains organizational data that is binding for the entire project.

Work Breakdown Structure (WBS)

WBS is a *hierarchical model* of the tasks to be performed in the project. It provides overview of the project and forms basis for the project organization and coordination. It shows work, time and money spent on the project. You can use it to plan dates & costs and allocate budget. The Work Breakdown Structure can be displayed according to:

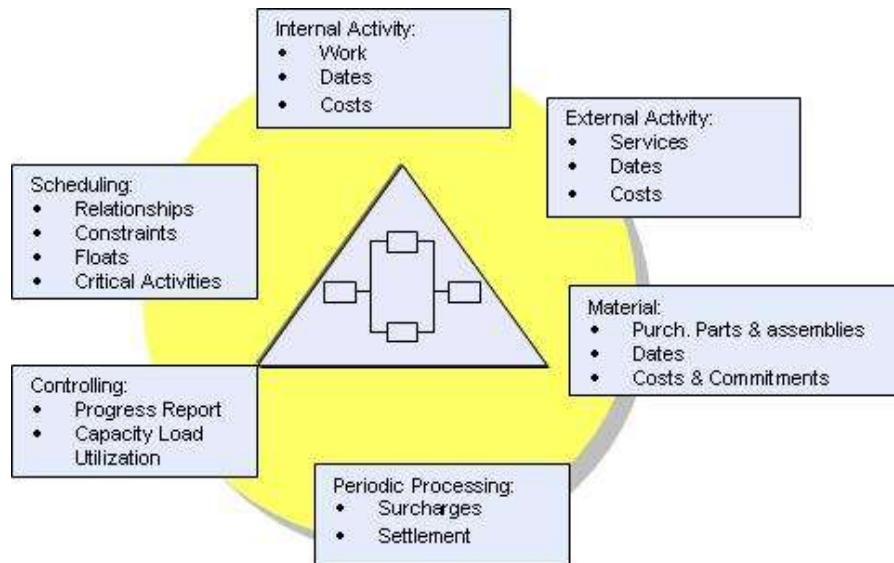
- Phase
- Function
- Object

Individual tasks required to carry out the project is called WBS elements.



Network

The network describes project processing. Thus the Network contains instructions on how to carry out activities in a specific way, in a specific order and in a specific time period. Main elements of network are activities and relationships. It forms the basis for planning, analysing and monitoring time schedules, costs, & resources.



Activities

Activities are used to plan project staffing, capacity, material, PRTs, and service requirements. When activities are assigned to WBS elements, detailed view of costs, dates, and capacities at WBS element could be obtained. It's a task in a network which has a defined start and finish. An activity can be broken down into activity elements. Three categories of activities in the Project System:

- Internal Activities
- External Activities
- General Cost Activities

Activity element

Activity element is an activity which is subordinate to another activity. Activity elements contain the same information as activities. Three categories of activity elements in the Project System:

- Work elements
- External Elements
- General Cost Elements

Activity Type

It's a physical measure of activity output of a cost centre .e.g. hours, number of units produced, machine times, production times.

Milestones

Milestones are the events in the project to which particular importance is attached or which trigger predefined functions. In general they indicate transition between different departments or phases. Milestones are assigned to activities and WBS elements. In PS milestones are used to:

- Trigger predefined functions in network activities.
- Carry out earned value analysis.
- Determine dates in billing plan for sales orders.

Relationships

You use relationships to depict chronological and technical dependencies between activities. The relationship determines the nature of the link between the individual activities. :

- **FS Relationship:** An activity does not start until the preceding activity is completed.
- **SS Relationship:** An activity cannot start unless another activity has started.
- **FF Relationship:** An activity cannot be completed until another activity has been completed.
- **SF Relationship:** An activity cannot be completed until another succeeding activity has started.

Confirmations

It is a part of network control. It documents the state of processing for the network activities and activity elements. There are two types of confirmations Partial & Final. Confirmations are used to record:

- The work centre where the activity was carried out.
- The person who carried out this activity.
- The yield and scrap produced in an activity.
- The actual values for the duration and dates.

Settlement

As a rule, projects are used to collect and monitor costs, but are not usually the final cost object. For this reason the costs in atypical project will be settled at the end of the period. To this end, you store settlement rules in the activity / WBS element requiring settlement. They contain information on settlement receivers, cost apportionment and control data. Settlement receivers could be cost centres, G/L accounts. Etc.

Budget

The budget is the approved cost structure for an action or project in a particular period. Budgeting differs from cost plan in that it is binding. In the approval phase you prescribe your project funds in the form of budget. It is possible to allocate overall and annual budget in parallel.

Work Centres

Work centre represents the resources responsible for executing an activity. In a work centre, you can enter the available capacity and an operating time. You can arrange the work centres in a hierarchy for capacity evaluation purposes. Internal activities are assigned to work centres (resources) to be completed.

Profit Centre

It is subdivision of business organization which is set up for internal management control purposes. Profit centres divide business up on a management basis. The basic aim of profit centre accounting is to present areas of the business as entities operating independently in the market.

Example instance of a construction project of a building.

Let say there is a construction company that is going to construct a Multi-Storey building.

The first thing that needs to be created in the system is ***Project Definition***. The entire activity plan for this project will come under this project definition. ***Overall budget*** needs to be estimated, assigned and get approved for the project. Project timeline needs to be decided. Project could be then subdivided into separate parts which are actually ***WBS elements*** lets say in this e.g. we have 4 main WBS elements:

1. Land Acquisition
2. Procurement
3. Construction
4. General

To every above WBS, budget and time line need to estimated.

Now to take an example we will drill down two WBS elements namely ***Land Acquisition and Construction***.

Land Acquisition could be sub divided into activities as in

- Generation of request document for the land.
- Soil testing
- Approval for construction
- Preparation of purchase order of the land

Above 4 activities should be done in sequence. As in second activity will start only after request document for the land is completed and so on. Hence these activities have **FS Relationship** between them.

– Now take WBS element **Construction**.

It could be sub divided in to floor wise tasks. Let say, we have four story building and each floor has 4 flats. So under WBS element '**Construction**' we may create separate **sub WBS element** for each floor. Each sub WBS element of floor may be divided further into **last level of WBS elements** one for each flat on the floor. This way we will have **hierarchy of the WBS elements**.

Finally last level of WBS elements i.e. WBS elements of the flats will have activities like

1. Plumbing
2. Electrical work
3. Flooring
4. Painting
5. Furniture

These activities form the Network.

Let say Plumbing and Electricity work are the **external activities** as they are given to a outside contractor. While the other activities are done by company labor so they are **internal activities**. Plumbing and Electrical work could be started simultaneously. While flooring needs be done once the plumbing and Electrical work is complete. Hence, they have FS relationship and so on.... Now if we talk about **milestones**:

1. First milestone could be purchasing the land
2. Second could construct the base of the building.
3. Third could be construct the floors
4. Fourth could be completion of plumbing and electrical work for all the flats. And so on....

2.4 SAP QM (Quality Management)

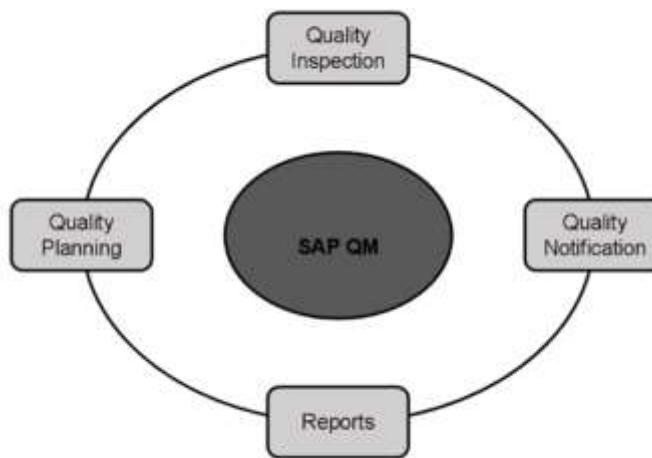
SAP Quality Management is a part of SAP R/3 system and is integrated with other SAP modules like SAP Material Management (MM), Production Planning (PP), and Plant Maintenance (PM). QM is an integral part of logistic management and it is used to perform quality functions such as quality planning, quality assurance, and quality control, at various stages such as incoming material stage, in-process manufacturing process stage, and after production as well.

With Quality Management module, you can implement the key modules of QM system as defined in manufacturing standards like ISO 9000.

As SAP Quality Management is an integral part of SAP R/3 system, it performs the following key functions in manufacturing of goods-

- **Quality Planning:** Quality planning allows to plan the inspection of goods from the vendor, raw material, work-in-process, and final product
- **Quality Notifications:** Quality notification includes the defect identification and steps to be taken by quality department
- **Quality Inspection:** Here, quality results are captured and decision is taken as to whether an inspection lot is to be accepted or rejected.

In the following diagram, you can see the key components that are involved in SAP Quality Management process.



Quality Planning

Quality planning includes data for quality planning and how the quality process has to be performed?

You perform Quality inspection plan. It is used to define the way you can inspect an item and the steps involved to perform an inspection. It also determines the characteristics of an item to be inspected and what equipment are required to perform an inspection.

Inspection plan definition is an important part of the QM planning process. The inspection plan contains the number of characteristics of the item to be inspected and the list of tests to be performed for performing the inspection.

Inspection planning can be done for raw material, work in progress and finished products.

Quality Assurance

Under Quality assurance, it includes the quality inspection. A Quality inspection involves someone from the quality department inspecting an item as per the defined points in inspection plan. You perform the inspection based on one or more inspection lots, where a lot is a request to inspect a specific item.

Quality Control

Under Quality Control, you have quality notifications, standard reports, and Quality notification system. Quality control determine what actions need to be taken as after defects are detected.

The quality notification process includes recording problem that is either identified by a customer for a product manufactured in an organization, or in a company against the product of a supplier/vendor. Quality notification can be raised internally to raise an issue that have arisen on the production line.

Standard Reports

You can create reports in QM system to check how many times a product has been identified with a defect and improvement areas that your company has to implement. There are number of reports that can be generated in QM-

Material Defects Report

It can be used to check the number of times an item has been identified with a defect status.

Vendor Defect Report

This report is used to show the number of defected material supplied by a vendor. This can be checked by examining the inspection lots of the goods received. A quality department can highlight vendors who have supplied material which are failing in inspection.

Customer Defect Report

This report is used to show the defects that were found on inspections for outbound deliveries. This helps the organization to improve the quality of goods delivered and hence raise the customer satisfaction.

SAP QM – Functions

These functions comprise to form Quality Management process. In SAP QM system, you can perform the following functions-

- Quality Planning
- Quality Inspection
- Quality Control
- Quality Certificates

Unit 3: Introduction to SAP Technical ERP Module

Learning Outcomes:

- Basic Understanding of SAP Basis and SAP Security Module
- Understanding different features in SAP HANA
- Concept of SAP CRM
- Introduction to SAP ABAP

3.1 SAP BASIS (Business Application Software Integrated Solution)

SAP BASIS stands for **Business Application Software Integrated Solution**. It is a set of tools that will work as a **bridge between your operating system, communication protocols, the various business applications, and database**. SAP BASIS also includes different administration activities.

These are **load balancing, installation**, and also maintaining the performance of the SAP systems that are being executed on SAP ABAP or Java stack. You can say that BASIS is an operating system for ABAP and SAP applications.

The **BASIS administrators** have the responsibility of handling the system errors. They also keep a check on the SAP enterprise and cloud applications. In this article, we will delve deeper into the details of the technology.

SAP BASIS is required for handling all the technical layers of the **SAP stack**. It is a system administration platform that is required for handling SAP environments such as **SAP HANA**. Its main aim is to smoothly execute all the SAP systems in this environment.

The different functionalities of the SAP BASIS software are mentioned below:

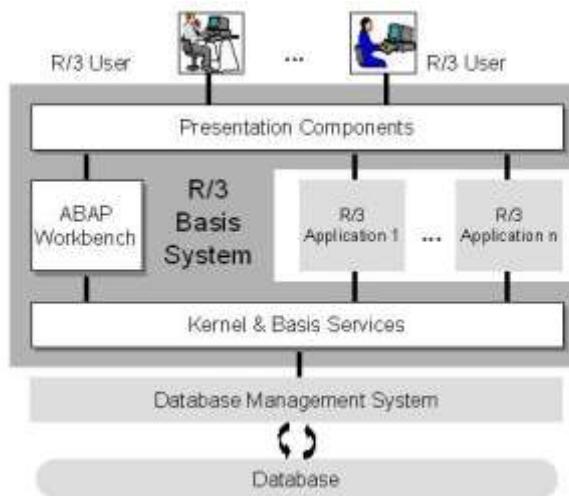
- Configuration and installation of SAP applications and systems
- Determining bugs, errors and tweaking the SAP systems for maximum performance
- Overseeing that the different users of the SAP systems have the correct authorization rights. They must be able to use the functions they need
- Executing and maintaining the background jobs
- Restoring the essential data and keeping backups
- It is used for configuring the SAP TMS. This is used for managing the transports between one or more SAP systems

- Installation, configuration, and maintenance of devices such as printers
- It provides you with other services such as communication between databases, memory management, exchange of business data, collection of web requests and application data etc.

You can consider SAP BASIS as the backbone of your entire SAP landscape.

The layers of the SAP BASIS system are:

- Presentation layer
- Application layer
- Database layer



Basis Administration

BASIS administration refers to all the different activities performed by an administrator. The aim is to keep the SAP environment stable.

This means **analyzing the system logs** and **determining errors**. Administration helps in preventing major system issues. For example, some business objects in a SAP system may be modified. In this case, SAP will initiate the locks that might temporarily restrict access. An SAP administrator has the ability to release this lock.

BASIS administration is also important for the following:

Maintenance

This includes reviewing system logs and records, identifying errors and fixing them. All this is done to make sure that your system is stable.

System tasks

BASIS administrators have the responsibility of planning and maintaining your system upgrades. Migrations are also handled. The admins take care of the transport

management and test the software updates. This is done to ensure compatibility with the SAP systems. They also check whether the updates are installed in the correct sequence.

Planning and scheduling

BASIS administration deals with scheduling the various system jobs. This is because many jobs run in your system background and consume resources. So, these resources need to be properly scheduled. It is important for keeping the system performance unaffected.

Moreover, the BASIS admins have to make the necessary adjustments to the SAP system to meet user's needs.

SAP Basis consultant's responsibilities

A SAP BASIS consultant runs a SAP landscape. He or she must have a strong technical background. Knowledge in subjects such as **UNIX, LINUX, Windows, MySQL, Oracle and Java is desirable**. Strong **hardware and networking skills** are also required. A consultant must have the ability to analyse problems. He or she must have communication skills to act according to customer demands.

The different roles and responsibilities of a SAP BASIS consultant are as follows:

- Monitoring system tasks and performance
- Installation, configuration and maintenance of SAP systems
- Database operations such as backup, schedule, maintenance and restoration
- Handling and maintaining SAP licenses
- Monitoring servers, background jobs scheduling and job deletions
- Creation of user profiles, providing them roles, locking/ unlocking these roles
- Maintenance of operation and profile modes
- Installation of R/3, Solution manager, Netweaver and Netweaver components
- Installing system upgrades, add-ons and support patches
- Initiating and halting the R/ 3 system
- Spooling and printing
- SAP R/ 3 router installation
- Operating, implementing and removing errors from **SNOTE**

Requirements for SAP BASIS

The primary objective or requirement of SAP BASIS is to keep the **SAP systems running smoothly, securely** and to ensure their **stability**. Both production and non-production environments are covered by Requirements for SAP BASIS. **Performance analysis, installation of patches, upgrades** and modifying parameters are all covered. Requirements also focus on troubleshooting and monitoring the systems regularly.

For all these day-to-day activities, a dedicated SAP Basis team is required.

Here are some reasons why such a team will be beneficial for any business:

Cost and time efficiency

Suppose you are facing problems while SAP Basis administration. The first reasonable step is to call the IT staff to handle these issues. But they might have other tasks and handling SAP Basis might be problematic.

This will be time-consuming and the operational costs will also be high. As these SAP systems need attention all day, a dedicated team will be very useful. Alternatively, you can also outsource the SAP Basis work to a professional team. A dedicated team or an outsourced team will be economical.

Dealing with complex problems easily

Your company may have inexperienced IT staff or junior SAP Basis employees. They will not be able to handle complicated SAP Basis complications. Even if they do, it takes a lot of time to fix. A dedicated team has certified and experienced professionals can handle SAP errors.

This way you can avoid mistakes that might affect system performance. The ecosystem of your enterprise may be complicated. In that case, a professional team will be beneficial for your company.

Meeting customer requirements

A dedicated and professional team of SAP Basis admins are better equipped at understanding the business requirements. They are able to communicate with other members of the IT team, managers and customers. So, all the intended customer services and support are delivered efficiently.

SAP Basis and Upgrades

When a SAP system is undergoing migration and upgrades, the SAP Basis professionals take the lead. This is because while migrating the business data, the former system also needs to be running. During the installation and configuration, the old system will continue generating business data. This data has to be migrated too. So, SAP Basis professionals will set up the entire SAP landscape with ease.

Before going for an upgrade, you have to plan it carefully. This is because the upgrade process may get complex and time-taking. Your aim must be to reduce the downtime and make it as effective as possible.

Some of the important concepts that you must be familiar with for SAP Basis upgrades are:

Software Update Manager (SUM)

This is a tool that is used for system maintenance, releasing upgrades and database migration. You can install SAP enhancement packages and convert the system to SAP S/4HANA. It comes with a Software Logistics Toolset 1.0. Patches are released for providing the latest features and bug fixes.

Upgrade guides

The Master Upgrade Guide is another important document. It has all the required specifications for upgrading your SAP system. You must read it for starting the upgrade as per your product version.

SAP Notes

The update of your system will require additional information. This might not be present in the upgrade guides. You will find this information within a range of SAP Notes. It is present in the SAP Support Portal. Before starting the process of upgrading the system, access the following documentation:

- The SAP Note for your database
- Software Logistics Toolset
- SAP Notes for DMO, if you are using the option of Database Migration
- To get the best results, use the latest SAP Notes. They are updated frequently.

We now have a basic understanding of SAP Basis. Let us compare it with another important software - SAP HANA.

SAP BASIS vs. SAP HANA

SAP Basis is used for holding the **SAP landscape and system together**. SAP HANA is a **relational database management system** used developed by SAP. Administration of SAP Hana will require a specific skill set. This can be acquired from SAP Basis administration. However, both are entirely different technologies. The main difference lies in their administration.

3.2 SAP Security

SAP security is a technical module that works within SAP systems to allow access where it's needed and prevent access where it's not. Establishing good internal security and access processes is a vital part of helping ensure your SAP system is protected and will function well. Protecting against external threats is important, but internal threats should not be underestimated. Make sure you manage them!

In a SAP Distributed Environment, there is always a need that you protect your critical information and data from unauthorized access. Human Errors, Incorrect Access Provisioning shouldn't allow unauthorized access to any system and there is a need to maintain and review the profile policies and system security policies in your SAP Environment.

To make the system secure, you should have good understanding of user access profiles, password policies, data encryption and authorization methods to be used in the system. You should regularly check SAP System Landscape and monitor all the changes that are made in configuration and access profiles.

The standard super users should be well-protected and user profile parameters and values should be set carefully to meet the system security requirements. While communicating over a network, you should understand the network topology and network services should be reviewed and enabled after considerable checks. Data over the network should be well protected by using private keys.



Why Is SAP Security Important?

SAP systems store large amounts of confidential or sensitive data. Users on your network using an SAP system must have access to everything they need to do their jobs; at the same time, they should not have access to important data, such as financial records or confidential information. **If an employee accidentally accesses data that should be restricted, they could cause problems by deleting or moving something.** An even worse scenario is if someone accesses sensitive data deliberately, whether to damage your business, leak data or commit fraud. Moreover, for compliance reasons, in some industries (like those involving health or financial data), certain types of information must be carefully protected.

How Does SAP Security Work?

The SAP ERP offerings include software dealing with goods and services, sales, finance, accounting, human resources, manufacturing, and logistics. All businesses need systems that are interconnected with the ability to share information between different parts of the business as needed. ERP systems integrate back-office functions, like organizational plans, data analysis, stock management, and governance planning—as well as front-office functions, including customer relationship management (CRM) and e-business.

ERP is comprised of several applications, including those relating to human resources, accounting, CRM, sales, and so on. By integrating these processes and centralizing their management, **you can save time and money.**

Once the SAP system is set up, SAP security is there to help ensure that the system works as intended, without any problems with security or data access. There are three areas that SAP security deals with:

- **Confidentiality.** This means no data should be disclosed in an unauthorized manner.
- **Integrity.** No data should be modified in an unauthorized way.
- **Availability.** Distributed denial-of-service (DDoS) attacks should not occur.

Why You Need SAP Security



Authentication Mechanism in a SAP System

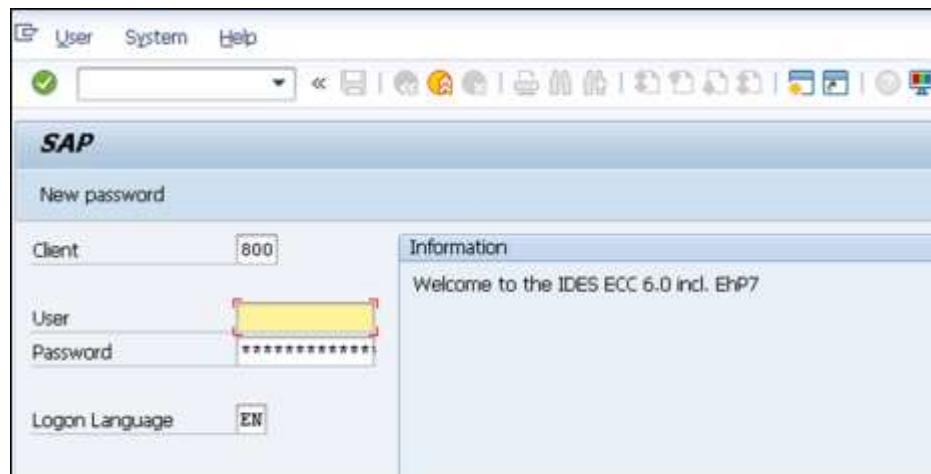
Authentication mechanism defines the way you access your SAP system. There are various authentication methods that are provided:

- User Id's and user management tools
- Secure Network Communication
- SAP Logon Tickets
- X.509 Client Certificates

User ID's and User Management Tools

Most common method of authentication in an SAP system is by using the username and password to login. The User ID's to login are created by the SAP Administrator. To provide secure authentication mechanism via the username and password, there is a need to define password policies that does not allow users to set easy predicted password.

SAP provides various default parameters that you should set to define password policies password length, password complexity, default password change, etc.



3.3 SAP HANA (High Performance Analytic Appliance)

SAP HANA is a combination of HANA Database, Data Modeling, HANA Administration and Data Provisioning in one single suite. In SAP HANA, HANA stands for High-Performance Analytic Appliance.

Features of SAP HANA

The main features of SAP HANA are given below –

- SAP HANA is a combination of software and hardware innovation to process huge amounts of real time data.
- Based on multi core architecture in distributed system environment.
- Based on row and column type of data-storage in database.
- Used extensively in Memory Computing Engine (IMCE) to process and analyze massive amounts of real time data.
- It reduces cost of ownership, increases application performance, enable new applications to run-on real-time environment that were not possible before.

Need for SAP HANA

Today, most successful companies respond quickly to market changes and new opportunities. A key to this is the effective and efficient use of data and information by analysts and managers.

HANA overcomes the limitations mentioned below –

- Due to the increase in “Data Volume”, it is a challenge for the companies to provide access to real time data for analysis and business use.
- It involves high maintenance costs for IT companies to store and maintain large data volumes.
- Due to unavailability of real time data, analysis and processing results are delayed.

SAP HANA Vendors

SAP has partnered with leading IT hardware vendors like IBM, Dell, Cisco etc. and combined it with SAP licensed services and technology to sell SAP HANA platform.

Top few Vendors include –

- IBM
- Dell
- HP
- Cisco
- Fujitsu
- Lenovo (China)
- NEC
- Huawei

SAP HANA Installation

HANA Hardware vendors provide preconfigured appliances for hardware, Operating System and SAP software product.

Vendor finalizes the installation by an onsite setup and configuration of HANA components. This onsite visit includes deployment of HANA system in Data Center, Connectivity to Organization Network, SAP system ID adaptation, updates from Solution Manager, SAP Router Connectivity, SSL Enablement and other system configuration.

In-Memory Computing Engine

An In-Memory database means all the data from the source system is stored in a RAM memory. In a conventional Database system, all data is stored in hard disk. SAP HANA In-Memory Database wastes no time in loading the data from hard disk to RAM. It provides faster access of data to multicore CPUs for information processing and analysis.

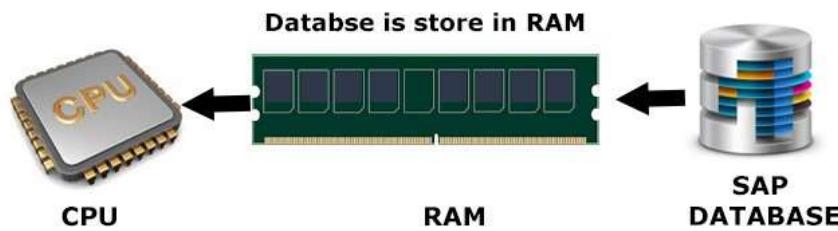
Features of In-Memory Database

The main features of SAP HANA in-memory database are –

- SAP HANA is Hybrid In-memory database.
- It combines row based, column based and Object-Oriented base technology.
- It uses parallel processing with multicore CPU Architecture.
- Conventional Database reads memory data in 5 milliseconds. SAP HANA In-Memory database reads data in 5 nanoseconds.

It means, memory reads in HANA database are 1 million times faster than a conventional database hard disk memory reads.

SAP HANA In-Memory Computing



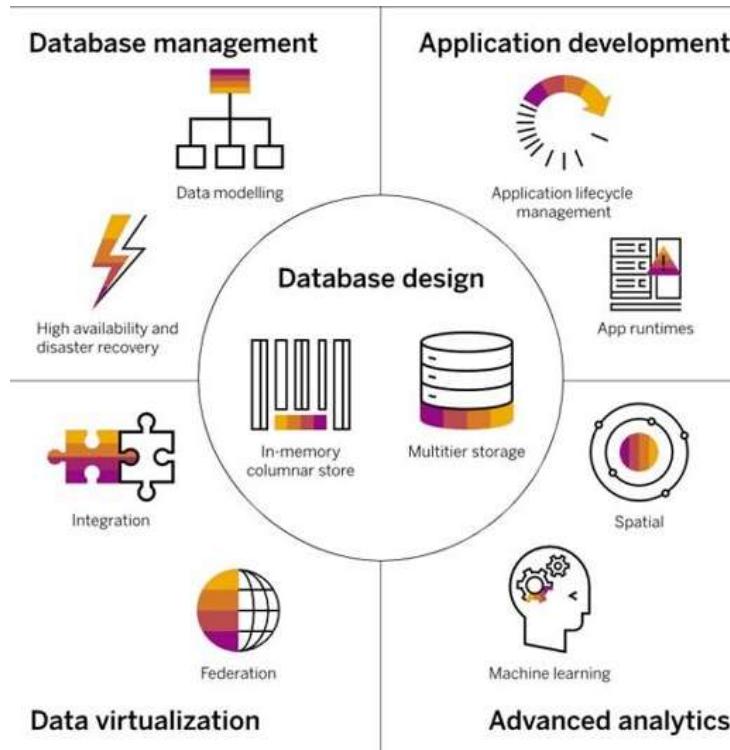
Analysts want to see current data immediately in real time and do not want to wait for data until it is loaded to SAP BW system. SAP HANA In-Memory processing allows loading of real time data with the use of various data provisioning techniques.

Advantages of In-Memory Database

- HANA database takes advantage of in-memory processing to deliver the fastest data-retrieval speeds, which is enticing to companies struggling with high-scale online transactions or timely forecasting and planning.
- Disk-based storage is still the enterprise standard and price of RAM has been declining steadily, so memory-intensive architectures will eventually replace slow, mechanical spinning disks and will lower the cost of data storage.
- In-Memory Column-based storage provides data compression up to 11 times, thus, reducing the storage space of huge data.

HANA Architecture

SAP HANA's in-memory, column-oriented architecture is built for fast queries and high-speed transactions – but it also includes – database management, application development, advanced analytical processing, and flexible data virtualization.



<https://www.sap.com/india/products/hana.html>

Studio & Administrative View

SAP HANA studio is an Eclipse-based tool. SAP HANA studio is both the central development environment and the main administration tool for HANA system. Additional features are –

- It is a client tool, which can be used to access local or remote HANA systems.
- It provides an environment for HANA Administration, HANA Information Modeling and Data Provisioning in HANA database.

SAP HANA Studio can be used on the following platforms –

- Microsoft Windows 32- and 64-bit versions of: Windows XP, Windows Vista, Windows 7
- SUSE Linux Enterprise Server SLES11: x86 64 bit
- Mac OS, HANA studio client is not available

SAP HANA Studio Perspectives / Features

SAP HANA Studio provides perspectives to work on the following HANA features. You can choose Perspective in HANA Studio from the following option –

HANA Studio → Window → Open Perspective → Other



Sap Hana Studio Administration

Toolset for various administration tasks, excluding transportable design-time repository objects. General troubleshooting tools like tracing, the catalog browser and SQL Console are also included.

SAP HANA Studio Application Development

By default, all features are installed.

To Perform HANA Database Administration and monitoring features, SAP HANA Administration Console Perspective can be used.

Administrator Editor can be accessed in several ways –

- From System View Toolbar – Choose Open Administration default button
- In System View – Double Click on HANA System or Open Perspective

HANA Studio: Administrator Editor

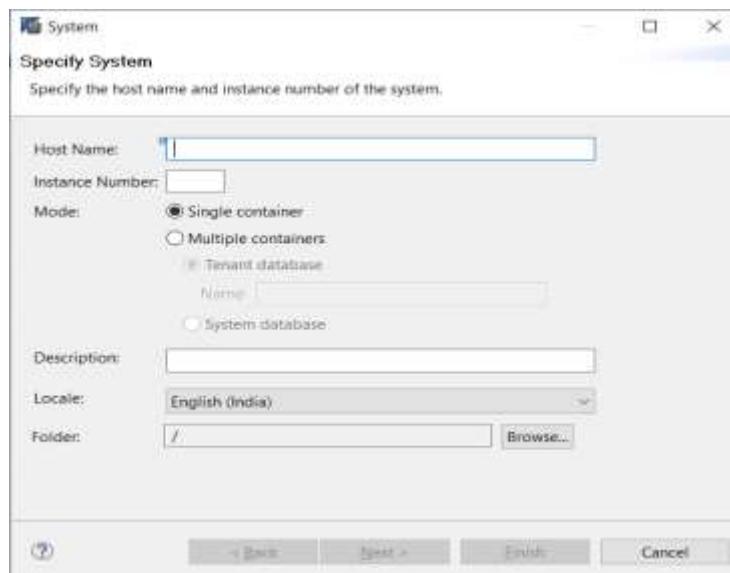
In Administration View: HANA studio provides multiple tabs to check configuration and health of the HANA system.

Adding a HANA System to Studio

To add new HANA system, host name, instance number and database user name and password is required.

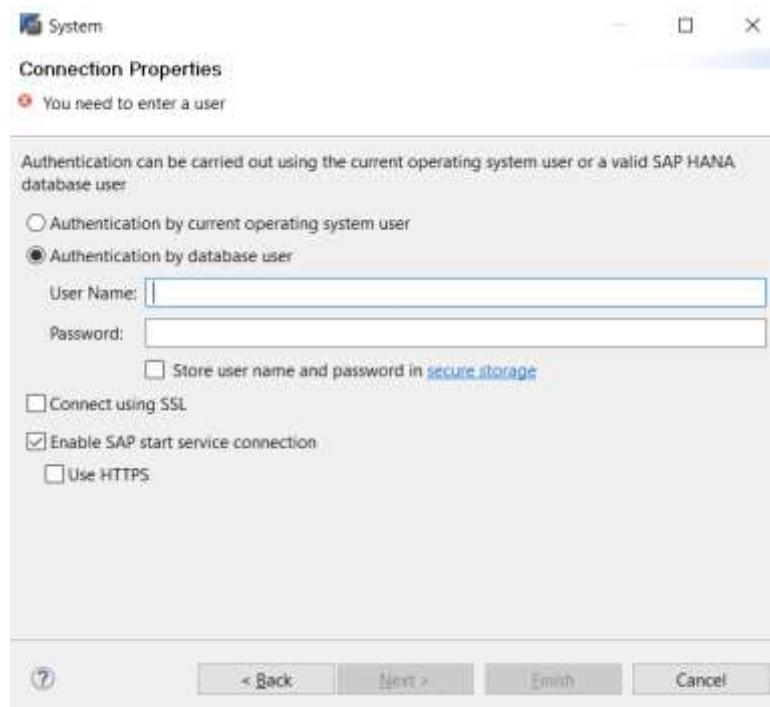
- Port 3615 should be open to connect to Database
- Port 31015 Instance No 10
- Port 30015 Instance No 00
- SSh port should also be opened

Adding a System to Hana Studio



To add a system to HANA studio, follow the given steps.

Right Click in Navigator space and click on Add System. Enter HANA system details, i.e., Host name & Instance number and click next. Enter Database user name and password to connect to SAP HANA database. Click on Next and then Finish.



Once you click on Finish, HANA system will be added to System View for administration and modeling purpose. Each HANA system has two main sub-nodes, Catalog and Content.

Catalog and Content

Catalog

It contains all available Schemas i.e., all data structures, tables and data, Column views, Procedures that can be used in Content tab.

Content

The Content tab contains a design time repository, which holds all information of data models created with the HANA Modeler. These models are organized in Packages. The content node provides different views on the same physical data.

System Monitor

System Monitor in HANA studio provides an overview of all your HANA system at a glance. From System Monitor, you can drill down into details of an individual system in Administration Editor. It talks about Data Disk, Log disk, Trace Disk, Alerts on resource usage with priority.

The following Information is available in System Monitor –

Column	Description
System ID	ID assigned to system when added
Operational State	Overall system status
Alerts	The system issues alerts when resource usage and statistical thresholds are violated. These alerts are categorized as low, medium, or high priority. There are also information alerts. The number of alerts and their status is shown here.
Data Disk (GB)	Size of the data volume on disk
Log Disk (GB)	Size of the log volume on disk
Trace Disk (GB)	Size of trace files on disk
Database Resident Memory (GB)	Size of resident memory at operating system level owing to SAP HANA database processes
System Resident Memory (GB)	Total size of resident memory in the operating system
Used Memory (GB)	Amount of physical memory used by the SAP HANA database
CPU (%)	Percentage of CPU used by the SAP HANA database
Hostname	Name of the server hosting the SAP HANA database
Instance Number	Instance number is the administrative unit that comprises the server software components
System Data Disk (GB)	Total disk space occupied on disk(s) containing data
System Log Disk (GB)	Total disk space occupied on disk(s) containing log files
Column	Description
System Trace Disk (GB)	Total disk space occupied on disk(s) containing trace files
System Physical Memory (GB)	Total amount of physical memory used
System CPU (%)	Overall CPU usage
Distributed	Indicates whether the system is running on a single host or it is a distributed system running on more than one host
Start Time First	Time that the first service started This value is updated when system is restarted for any reason.
Start Time Latest	Time that the last service was started, if, for example, one of the services was restarted individually
Version	Software version number of the SAP HANA studio
Platform	Operating system on which the SAP HANA studio is running
Number of Crash Dump Files	The number of crash dump files in the trace directory of the system

Information Modeler

SAP HANA Information Modeler; also known as HANA Data Modeler is heart of HANA System. It enables to create modeling views at the top of database tables and implement business logic to create a meaningful report for analysis.

Features of Information Modeler

- Provides multiple views of transactional data stored in physical tables of HANA database for analysis and business logic purpose.
- Informational modeler only works for column-based storage tables.
- Information Modeling Views are consumed by Java or HTML based applications or SAP tools like SAP Lumira or Analysis Office for reporting purposes.
- Also, it is possible to use third party tools like MS Excel to connect to HANA and create reports.
- SAP HANA Modeling Views exploit the real power of SAP HANA.

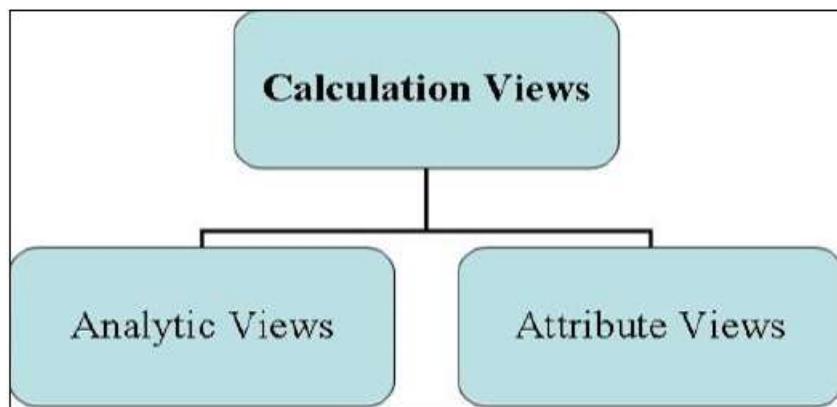
Information Modeling Views

Attribute View

Attributes are non-measurable elements in a database table. They represent master data and are similar to characteristics of BW. Attribute Views are dimensions in a database or are used to join dimensions or other attribute views in modeling.

Important features are –

- Attribute views are used in Analytic and Calculation views.
- Attribute view represents master data.
- Used to filter size of dimension tables in Analytic and Calculation View.



Analytic View

Analytic Views use the power of SAP HANA to perform calculations and aggregation functions on the tables in database. It has at least one fact table that has measures and primary keys of dimension tables and surrounded by dimension tables contain master data.

Important features are –

- Analytic views are designed to perform Star schema queries.
- Analytic views contain at least one fact table and multiple dimension tables with master data and perform calculations and aggregations
- They are similar to Info Cubes and Info objects in SAP BW.

- Analytic views can be created on top of Attribute views and Fact tables and perform calculations like number of unit's sold, total price, etc.

Calculation Views

Calculation Views are used on top of Analytic and Attribute views to perform complex calculations, which are not possible with Analytic Views. Calculation view is a combination of base column tables, Attribute views and Analytic views to provide business logic.

Important features are –

- Calculation Views are defined either graphical using HANA Modeling feature or scripted in SQL.
- It is created to perform complex calculations, which are not possible with other views- Attribute and Analytic views of SAP HANA modeler.

Difference between SAP HANA, S/4 HANA & C/4 HANA

SAP HANA

SAP HANA is an in-memory database technology that runs the SAP Landscape serving as the backend. SAP HANA is a column-oriented relational database management system (RDBMS) that serves a computing platform for many SAP applications. SAP HANA is primarily used as a core technology or database platform in systems or environments involving business operations related to sales, HR, finance, logistics, etc.

Its in-memory technology, column-based storage, OLAP and OLTP support makes data storing, retrieving, processing and analyzing 10 times faster and storage efficient than any other traditional RDBMS.

In addition to this, HANA uses main memory efficiently and leaves significantly less memory footprint than traditional systems. It uses compression techniques on data to reduce the space taken by data in column stores.

SAP S/4 HANA

Launched in February 2015, SAP S/4 HANA is SAP's next-generation business suite designed only to run on SAP HANA. **S/4 HANA stands for SAP Business Suite 4 SAP HANA.** SAP S/4 HANA is the fourth business suite version coming after SAP R/3.

Also, by making it solely compatible with SAP HANA system at its backend, SAP replaces the old SAP ECC/ERP system with SAP HANA. However, S/4 HANA is based around its successor i.e. ECC (ERP Control Center) solution and treats it as its core technology.

SAP S/4 HANA is a new **Enterprise Resource Planning (ERP) solution** with *simplified data, a simple tool design, agile, easier to use, perform complex calculations, and handling greater amounts of data*. It can be deployed on-premise, on-cloud or as a hybrid system.

S/4 HANA does not support batch processing for its data which makes retrieval and processing of data very fast and that too in real-time. Real-time analytics makes it possible to use S/4 HANA powered by SAP HANA to process and analyze data from the Internet of Things (IoT) or big data sources.

Key Difference between SAP HANA & SAP S/4 HANA

To understand the basic difference between the two, read the points below.

SAP HANA is an in-memory database technology which acts as the core technology for a lot of other SAP or non-SAP applications whereas SAP S/4 HANA is a new generation ERP solution which runs on SAP HANA database architecture.

S/4 HANA is a business suite launched as a robust ERP solution having both ERP and BI capabilities utilizing HANA's in-memory computing power. It is an in-memory version of the ERP Business Suite as it only runs on SAP HANA.

SAP C/4 HANA

SAP C/4HANA is a customer experience (CX) suite designed to change customer relationship management (CRM) as we currently know it. Today's CRM systems are focused on sales. However, customer relationships do not end once a sale is closed. Customer experience has become the top priority in boardrooms across the globe and with this shift in mindset and strategy, the legacy CRM systems no longer comply.

SAP C/4HANA is a "Next-generation CRM" because this solution modernized the legacy CRM solutions. SAP C/4HANA achieved it by encouraging positive customer experience at every stage of the customer journey, providing a consistent experience across all channels, and also real-time data.

The SAP Customer Experience suite, C/4HANA, includes 5 clouds that will cover and optimize all of your front-office operations.

- Marketing Cloud
- Sales Cloud
- Commerce Cloud
- Service Cloud
- Customer Data Cloud

Every cloud has multiple tools and advanced integrated technologies that will help you with your day-to-day operations.

With C/4HANA, companies will be able to automate time-consuming tasks, increase their workforce productivity and better understand their customers.

3.4 SAP CRM (Customer Relationship Management)

SAP CRM is the CRM tool provided by SAP and is used for many a business process

SAP CRM is a part of SAP business suite. It can implement customized business processes, integrate to other SAP and non-SAP systems, help achieve CRM strategies.

SAP CRM can help an organization to stay connected to customers. This way organization can achieve customer expectations with the types of services and products that he or she needs.

It also helps to achieve ‘Single face to customer’, which means the customer get regular & actual information independent of channel through which the he or she is contacting your company.

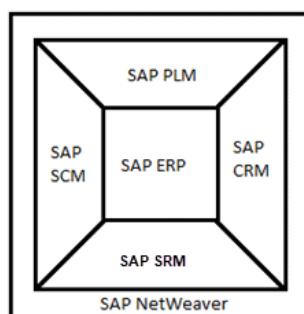
SAP CRM Overview

As a part of SAP Business Suite, SAP provides solutions which are flexible and open, and which support applications, databases, hardware platforms, & operating systems from most of the major vendors.

Following SAP solutions are the constituents of SAP Business Suite:

- SAP CRM – Customer Relationship Management
- SAP PLM – Product Lifecycle Management
- SAP SCM – Supply Chain Management
- SAP SRM –Supplier Relationship Management
- SAP ERP – Enterprise Resource Planning

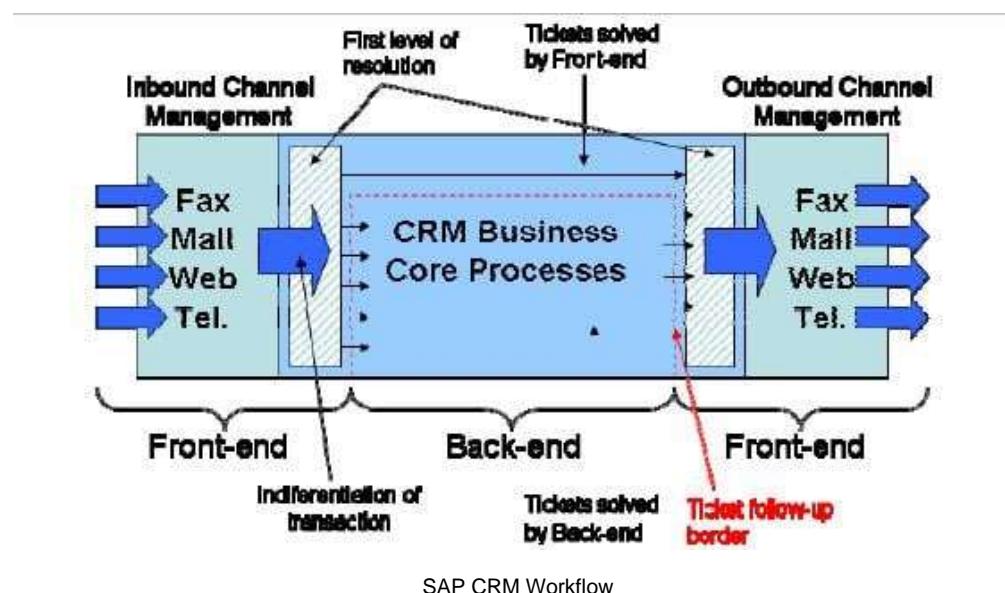
SAP Business Suite is based on SAP NetWeaver. NetWeaver provides the development and runtime environment for SAP applications and is used for the custom development and integration with other applications and systems.



SAP MM Module

SAP CRM is embedded in the business environment of the SAP Business Suite.

Typical SAP CRM WORK FLOW



A customer may raise an issue with the vendor via any medium like Fax, email , telephone etc. If the resolution can not be provided immediately by the front – end customer representatives they raise a ticket in SAP – CRM which is addressed by a more technically equipped personnel. The resolution is than forwarded to the customer.

Features of SAP CRM

- It is a part of SAP Business Suite to manage customer relationship.
- It supports all customer-focused business areas such as marketing, sales and service.
- CRM Analytics, a component of SAP CRM, enables your organization to gather all relevant information about various key factors such as a customer and analyze this knowledgebase to incorporate insights into operational processes and strategic decision-making.

SAP CRM Marketing



SAP CRM Marketing

- SAP CRM has provided extensive marketing functionalities
- It automates the marketing planning, campaign execution, & measurement of the marketing effort.
- SAP CRM unites the following key functions related to marketing on a user-friendly and configurable interface:

- Marketing Planning,
- Campaign Management,
- Lead Management,
- E-Marketing,
- Market Analytics,
- Customer Segmentation.

SAP CRM Sales

- SAP CRM is developed for handling customer contact anytime, anywhere.
- The companies can choose one or more of these SAP CRM Sales implementation:
- Telesales,
- Enterprise Sales,
- E-Selling and
- Field Sales.
- SAP CRM sales support the sales force of your business to be time efficient & effective in working.
- It provides information which leads an insight into action, & maintains focus on productive activity.
- Thus, SAP CRM Sales helps the sales force of your business to secure customers, and then to develop and maintain beneficial relationship with them.



SAP CRM Sales

- SAP CRM also provides aspect of sales forecasting and analytics that helps your business to collect historical & predictive information.
- It includes territory and account management which can be used to optimize & increase the effectiveness of your sales organization.
- It also includes Opportunity and pipeline management processes which provide maximum visibility in to the potential sales, sales processes, & methodologies which can lead to standardization of the company-specific best practices.
- It also provides seamless order to cash processes that enable your sales organization to manage the customer demands most effectively.

Thus, SAP CRM Sales have a lot of features like dedicated interactions, seamless integration, insightful information, always accessible, and is user-friendly.

SAP CRM Service

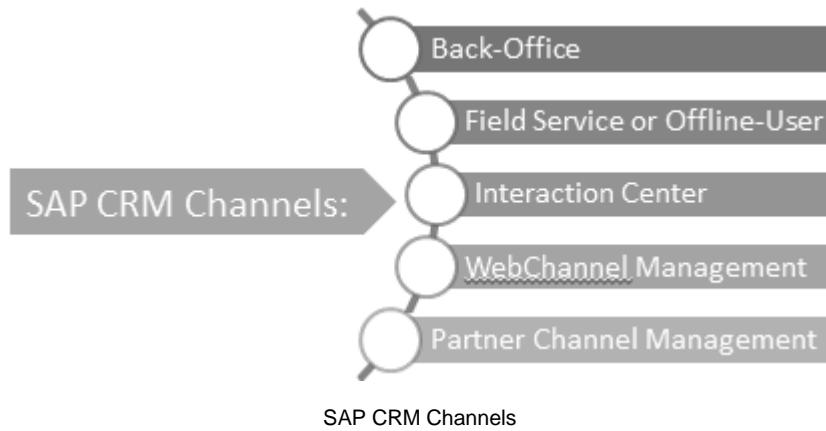


- All aspects related to the processing of the service order supported by SAP CRM service starts with responding to customer's initial inquiry till the confirmation & billing of the service provided to the customer.
- SAP CRM service also provides your organization with quotation creation & processing, creation of service order and assignment to field service representative.

SAP CRM Channels

SAP CRM provides implementations for different channels within your business such as Internet, telephony, field sales, and partners which leads to the optimization of your customer interactions. For all the different channels supported, SAP CRM provides your employee with an intuitive and user-friendly interface to carry out their daily work.

SAP CRM enables customers to implement different customer-specific requirements and industry-specific processes. For interaction with these implementations customers have different interaction channels offered by SAP CRM:

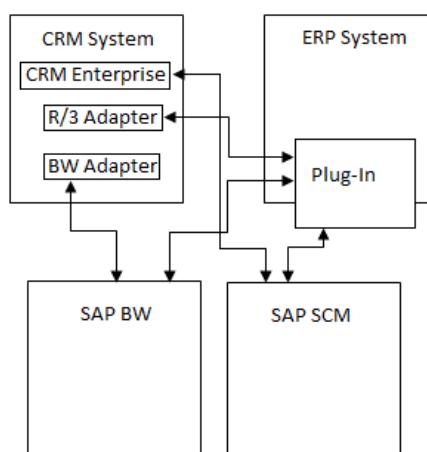


- **Back-office:** This is Role based web access. For each relevant task in the service, sales and marketing it represents the single point of entry. All relevant systems are assigned to a Business Role which is assigned to employees. Thus, an employee can get all the relevant systems into a single UI.

- **Field Service or Offline-User:** SAP CRM offers several field applications for field service representatives which they can access anywhere, anytime. These applications are developed for handheld devices.
- **Interaction Center:** SAP CRM provides the customer care employees with an interface which is comprehensively integrated with different communication channels like phone and E-mails. Also, it includes various features with which the employee can use while in communication with the customer for making note or working on the transaction itself.
- **WebChannel Management:** with this SAP CRM enables E-service, E-commerce, and E-marketing platform. These platforms are to provide personalized, reliable and convenient service to the target customers 24x7x365. This enables end customers to access & research data and with that as per requirement purchase services or products anytime, anywhere.
- **PCM – Partner Channel Management:** This interface is provided to support collaboration with resellers, dealers, agents etc. It combines the Web Channel Management with regular CRM to provide a complete solution for partner management.

Overview of SAP CRM Architecture

The SAP CRM solution incorporates the CRM components along with the SAP ERP, SAP SCM and SAP BI components. SAP CRM contains a central CRM system with access through various channels and a connection to other systems.



SAP CRM Architecture

Introduction to CRM WebClient User Interface

SAP CRM User Interface started with SAPGUI, and its growth has resulted in SAP CRM WebClient User Interface. CRM WebClient user interface is an enhanced version of the IC WebClient UI. Also, it is business role based UI; therefore, the content which will be visible to the user logged-in depends upon the business roles assigned to the user. This results in a simpler UI for the user, who will be able to access and process only those tasks which are relevant for him or her.



SAP CRM WebClient

CRM Web Client UI is component based software, which presents the CRM UI to the user in L-Shape. It contains Header in the top row and Navigation Bar on the left side, this constitutes the L-Shape. The remaining space on the CRM Web UI page is called Work Area. The Header area contains predefined system link like Log Off hyperlink.



SAP CRM WebClient User Interface

Following are the components of the Header area:

1. System Links
2. Saved Searches
3. Work Area Title
4. History

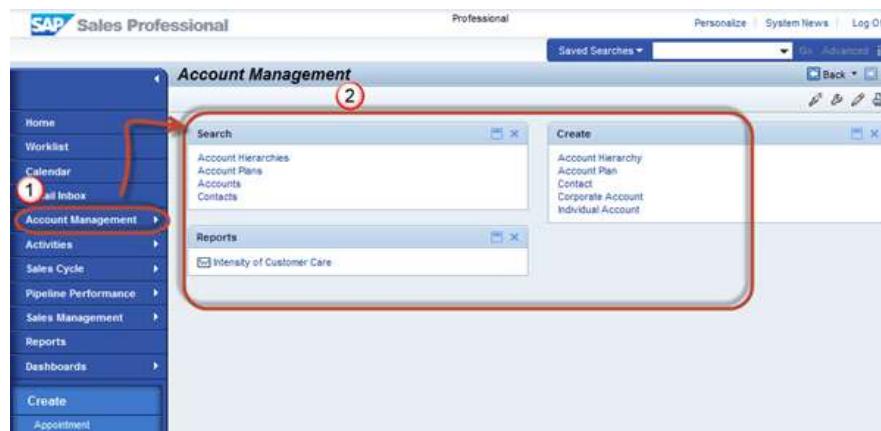


The position of the Header Area is fixed and cannot be changed. Navigation Bar contains links to various applications that are assigned to the logged-in user. The content of the Work Area gets updated with user action on the links available in Header area, Navigation bar or within the work area itself. The views are displayed in the CRM Web UI as Assignment Blocks. There are separate pages offered as implementation of the SAP CRM Web UI:

- Home page
- Worklist page
- Calendar
- E-Mail inbox
- Work Center
- Advanced Search Page
- Overview Page
- Assignment blocks

User can navigate between these pages using the links available in navigation bar, work centre or hyperlinks available in the search pages, applications or business transactions.

- As soon as a user logs in, he or she will be able to see the Home page.
- Further navigation to other pages or specific application can be accomplished with the navigation links in the Navigation Bar or in the work center.
- For example, user can access the below Work Centre for Account Management from the link available in the navigation bar:



- User can navigate to the below Account search using the link in the work center or using the Account Search link available in the second level of navigation bar:



3.5 SAP ABAP (Advanced Business Application Programming)

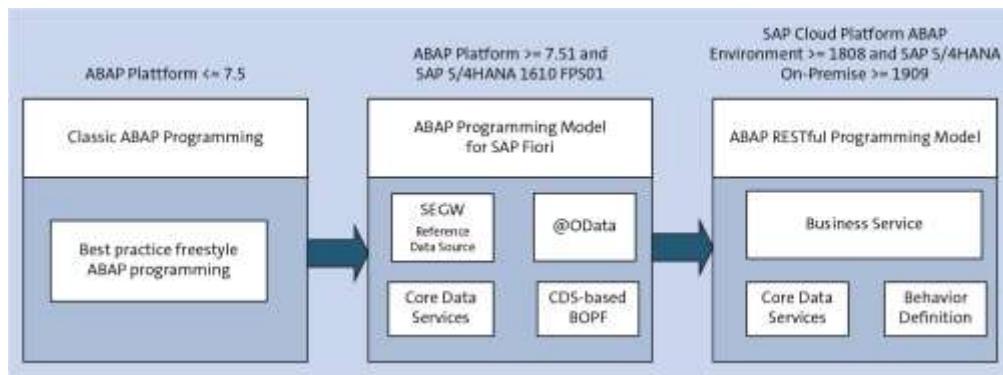
ABAP (Advanced Business Application Programming) is the name of SAP's proprietary, fourth-generation programming language. It was specifically developed to allow the mass-processing of data in SAP business applications.

ABAP is a multi-paradigm programming language, meaning programmers can utilize procedural, object-oriented, and other programming principles. While it is SAP's primary programming language, programs written with ABAP can run alongside those based on other programming languages such as Java, JavaScript, and SAPUI5.

A Brief History of ABAP

ABAP was first introduced by SAP in the 1980s. Throughout the years, various enhancements to the language increased what programmers could do with it. For example, through April 2000 programs could only be created *procedurally*, meaning a program had to follow a set of pre-defined "procedures" to perform a certain task successfully.

In May 2000, SAP changed ABAP with release 4.6C, allowing for object-oriented programming (OOP). This programming strategy involves multiple individual "objects" interacting with one another, allowing programs to grow more complex with the use of ABAP design patterns and other OOP practices.



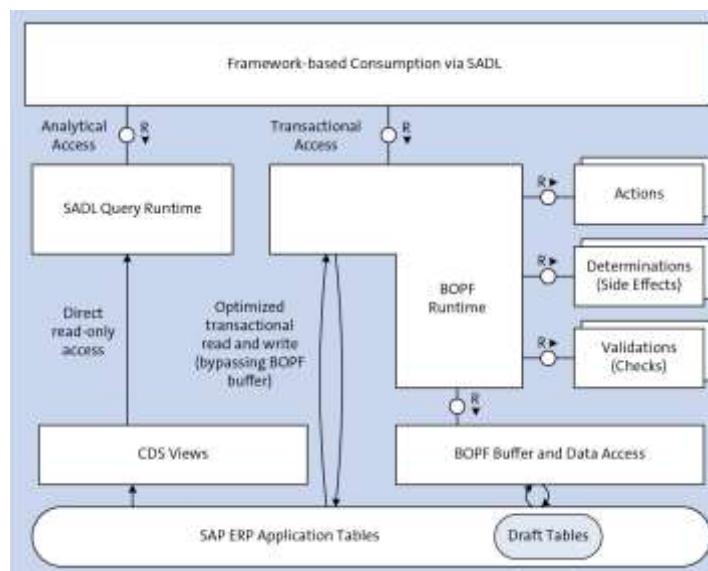
With the release of ABAP 7.4 and 7.5 in the early to mid-2010s, SAP gave object-oriented programmers using ABAP some powerful new features to play around with, vastly reducing the amount of code needed for common tasks.

Other new features made available to ABAP programmers in the 2010s were extended syntax for Open SQL, ABAP Managed Database Procedures (AMDP), and core data services (CDS) Views.

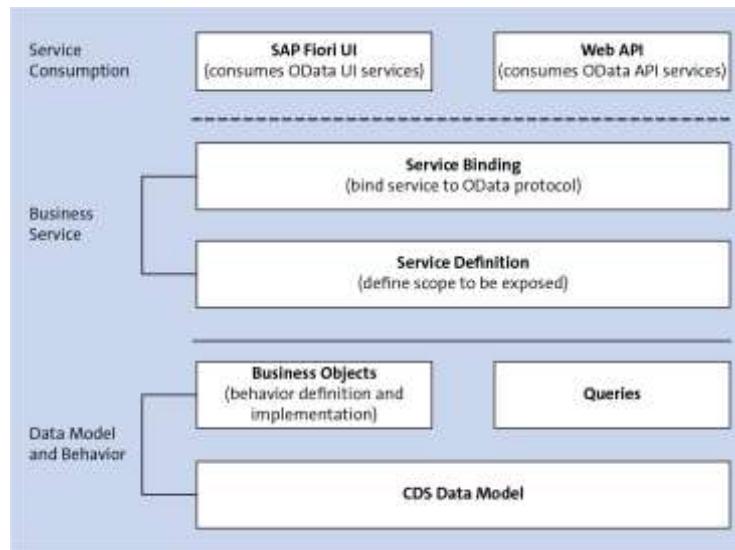
With the traditional, row-based database architecture of SAP R/3, it was important to have ABAP code run in the application layer rather than the database layer to save memory usage for further tasks.

During the early 2010s, many developers wondered if ABAP was to become increasingly obsolete as SAP acquired multiple cloud, non-ABAP-based solutions and pivoted existing products towards the cloud. But with the advent of SAP S/4HANA, and more importantly ABAP in the Cloud, the language was given new life, leading many to proclaim “ABAP’s not dead.”

These new platforms led to the creation of additional ABAP programming models. The first, the **ABAP programming model for SAP Fiori**, is used when developing SAP HANA-optimized OData services for SAP Fiori applications. These are based on core data services views and cover three application scenarios: analytical, transaction, and search.



The **ABAP RESTful programming model** is a very new paradigm based on the model for SAP S/4HANA, but eschews Business Object Processing Framework (BOPF) in place of a more advanced concept.



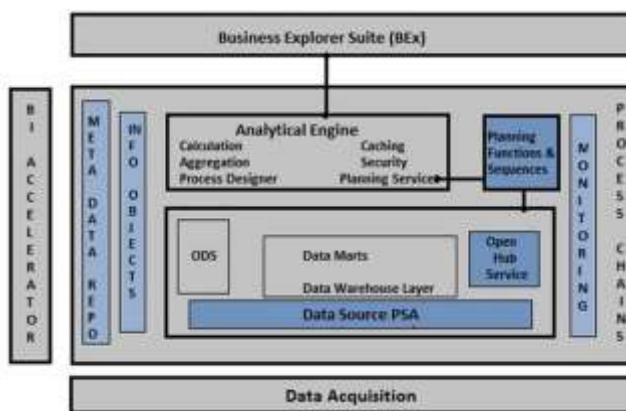
3.6 SAP BW (Business Warehouse)

SAP Business Intelligence (BI) means analyzing and reporting of data from different heterogeneous data sources. **SAP Business Warehouse (BW)** integrates data from different sources, transforms and consolidates the data, does data cleansing, and storing of data as well. It also includes data modeling, administration and staging area.

The data in SAP BW is managed with the help of a centralized tool known as SAP BI Administration Workbench. The BI platform provides infrastructure and functions which include –

- OLAP Processor
- Metadata Repository
- Process designer and other functions.

The following diagram shows an open, broad and standard based Architecture of Business Intelligence.



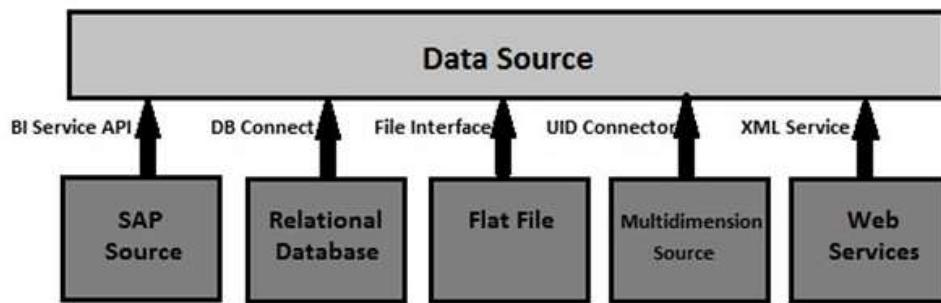
Reference: https://www.tutorialspoint.com/sap_bw/sap_bw_overview_of_sap_bi.htm

SAP BW is known as an open, standard tool which allows you to extract the data from different systems and then send it to the BI system.

In 1997, SAP had first introduced a product for reporting, analysis and data warehousing and it was named as **Business Warehouse Information System (BIW)**.

Data Acquisition in SAP BI

SAP BI allows you to acquire data from multiple data sources that can be distributed to different BI systems. A SAP Business Intelligence system can work as a target system for data transfer or source system for distribution of data to different BI targets.



Reference: https://www.tutorialspoint.com/sap_bw/sap_bw_overview_of_sap_bi.htm

As mentioned in the above image, you can see SAP BI source systems along with other systems –

- SAP systems (SAP Applications/SAP ECC)
- Relational Database (Oracle, SQL Server, etc.)
- Flat File (Excel, Notepad)
- Multidimensional Source systems (Universe using UDI connector)
- Web Services that transfer data to BI by means of push

When you go to SAP BI Administration workbench, the source system is defined there.

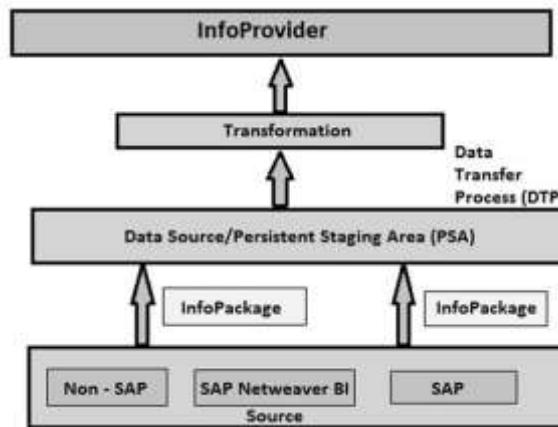
Go to RSA1 → Source Systems

You can load the data from any source in the data source structure into BI with an **InfoPackage**. Target system where the data is to be loaded is defined in the transformation.

InfoPackage

An **InfoPackage** is used to specify how and when to load data to the BI system from different data sources. An InfoPackage contains all the information on how the data is loaded from the source system to a data source or a PSA. InfoPackage consists of condition for requesting data from a source system.

BI Data Flow (InfoPackage and InfoProvider)



Reference: https://www.tutorialspoint.com/sap_bw/sap_bw_overview_of_sap_bi.htm

BI objects are divided into multiple BI content areas so that they can be used in an efficient way. This includes content area from all the key modules in an organization, which include –

- SCM
- CRM
- HR
- Finance Management

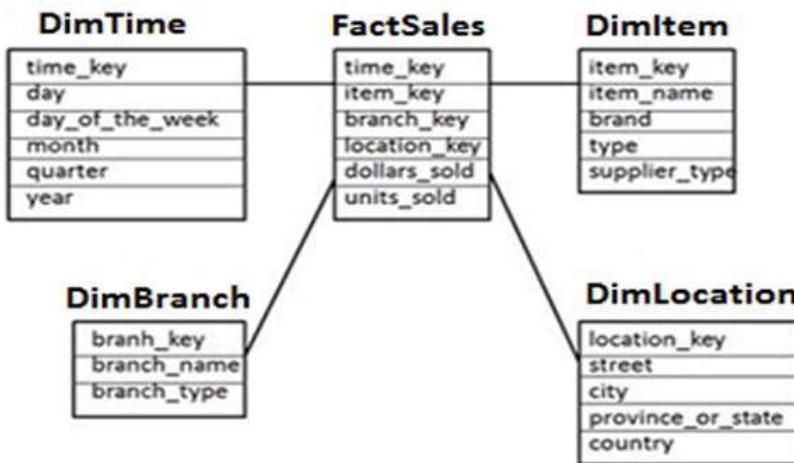
Data Warehousing

Star Schema

In Star Schema, each dimension is joined to one single fact table. Each dimension is represented by only one dimension and it is not further normalized. A dimension Table contains a set of attributes that are used to analyze the data.

For example – We have a fact table called FactSales that has primary keys for all the Dim tables and measures units_sold and dollars_sold to do analysis.

We have 4 Dimension tables – DimTime, DimItem, DimBranch, DimLocation as shown in the following image.



Reference: https://www.tutorialspoint.com/sap_bw/sap_bw_data_warehousing.htm

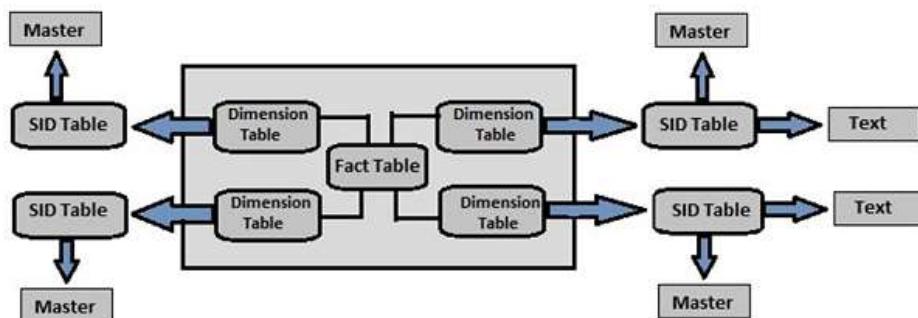
Each dimension table is connected to a fact table as the fact table has the primary Key for each dimension tables that are used to join two tables.

Extended Star Schema

In Extended Star schema, fact tables are connected to dimension tables and this dimension table is further connected to SID table and this SID table is connected to master data tables. In an extended star schema, you have the fact and dimension tables inside the cube, however SID tables are outside the cube.

In the extended star schema one fact table can connect to 16 dimension tables and each dimension table is assigned with 248 maximum SID tables. These SID tables are also called as characteristics and each characteristic can have master data tables like ATTR, Text, etc.

- **ATTR** – It is used to store all the attribute data.
- **Text** – It is used to store description in multiple languages.



Reference: https://www.tutorialspoint.com/sap_bw/sap_bw_data_warehousing.htm

InfoArea

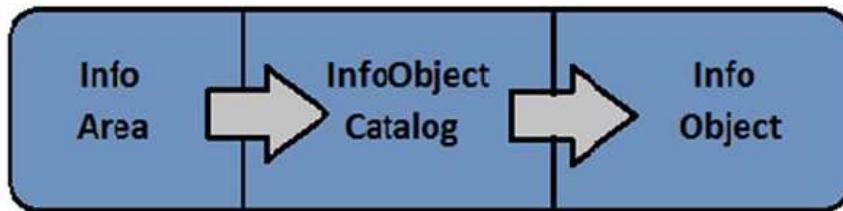
Info Area in SAP BI is used to group similar types of objects together. Info Area is used to manage Info Cubes and InfoObjects. Each InfoObjects resides in an Info Area and you can define it in a folder which is used to hold similar files together.

InfoObjects

InfoObjects are known as the smallest unit in SAP BI and are used in Info Providers, DSO's, Multi providers, etc. Each Info Provider contains multiple InfoObjects.

InfoObjects are used in reports to analyze the data stored and to provide information to decision makers. InfoObjects can be categorized into the following categories –

- Characteristics like Customer, Product, etc.
- Units like Quantity sold, currency, etc.
- Key Figures like Total Revenue, Profit, etc.
- Time characteristics like Year, quarter, etc.



Reference: https://www.tutorialspoint.com/sap_bw/sap_bw_data_warehousing.htm

Unit 4: ABAP Language Basics & Class

Learning Outcomes:

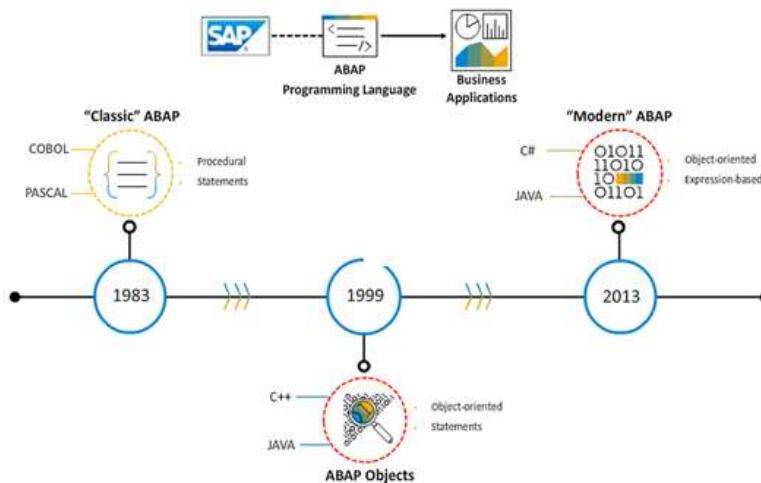
- Understanding the basic of SAP ABAP
- Basic Data Types used in ABAP
- Concept of Classes and Objects in SAP ABAP

4.1 Understanding the Basic feature of ABAP

Origin and Evolution of ABAP

ABAP is a programming language developed by SAP for the development of business applications in the SAP environment. ABAP is the fourth-generation programming language and technical module of SAP, which is used to develop the applications related to SAP. With the help of ABAP, one can customize SAP according to the requirement.

What is ABAP?



Reference: <https://learning.sap.com/>

- ABAP Stands for Advanced Business Application Programming. It is a high-level programming language, which is developed and maintained by the SAP AG Software Company for the development of SAP applications.
- ABAP is the core programming language that is used in SAP ERP software. Since it is the fourth-generation language, hence also known as ABAP/4.

- The ABAP was originally developed for generating the SAP R/2 reports. The SAP R/2 was used to enable the corporations to develop mainframe business applications, which are mainly used for financial accounting and material Management.

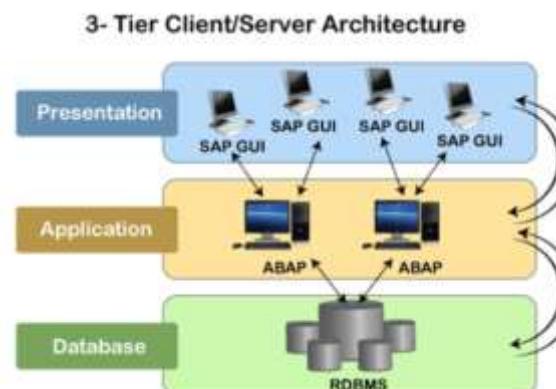
Why learn ABAP?

Below are some reasons that explain the reason to learn ABAP:

- ABAP is the core language used in the SAP R/3 platform.
- As a programming language, ABAP is easy to learn that provides both procedural and object-oriented concepts of programming.
- It is the main technology that ABAP developers work on to develop various SAP applications.
- ABAP provides various important features such as data sharing, exception handling, data persistence, enable to make enhancements, etc.
- Developers can use ABAP language to make any changes in applications of ERP systems

Process of ABAP in SAP kernel

We can understand the process of ABAP in SAP by using **3 tier architecture of SAP**.



Reference: <https://www.javatpoint.com/>

4.2 Basic Data Objects & Data Types

Elementary Data Types

ABAP offers the programmer a rich assortment of fixed length as well as variable length data types. Following table lists down ABAP elementary data types –

Type	Keyword
Byte field	X
Text field	C
Integer	I
Floating point	F
Packed number	P
Text string	STRING

Some of the fields and numbers can be modified using one or more names as the following –

- byte
- numeric
- character-like

The following table shows the data type, how much memory it takes to store the value in memory, and the minimum and maximum value that could be stored in such type of variables.

Type	Typical Length	Typical Range
X	1 byte	Any byte values (00 to FF)
C	1 character	1 to 65535

N (numeric text filed)	1 character	1 to 65535
D (character-like date)	8 characters	8 characters
T (character-like time)	6 characters	6 characters
I	4 bytes	-2147483648 to 2147483647
F	8 bytes	2.2250738585072014E-308 to 1.7976931348623157E+308 positive or negative
P	8 bytes	[$-10^{(2\text{len}-1)+1}$] to [$+10^{(2\text{len}-1)-1}$] (where len = fixed length)
STRING	Variable	Any alphanumeric characters
XSTRING (byte string)	Variable	Any byte values (00 to FF)

Complex & Reference Types

The complex types are classified into Structure types and Table types. In the structure types, elementary types and structures (i.e. structure embedded in a structure) are grouped together. You may consider only the grouping of elementary types.

The table types are better known as arrays in other programming languages. Arrays can be simple or structure arrays. In ABAP, arrays are called internal tables and they can be declared and operated upon in many ways when compared to other programming languages.

S.No.	Parameter & Description
1	Line or row type Row of an internal table can be of elementary, complex or reference type.
2	Key Specifies a field or a group of fields as a key of an internal table that identifies the table rows. A key contains the fields of elementary types.
3	Access method Describes how ABAP programs access individual table entries.

Variables

Variables are named data objects used to store values within the allotted memory area of a program. As the name suggests, users can change the content of variables with the help of ABAP statements. Each variable in ABAP has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

The basic form of a variable declaration is –

DATA <f> TYPE <type> VALUE <val>.

There are three kinds of variables in ABAP –

- Static Variables
- Reference Variables
- System Variables

Static Variables

- Static variables are declared in subroutines, function modules, and static methods.
- The lifetime is linked to the context of the declaration.
- With 'CLASS-DATA' statement, you can declare variables within the classes.

Reference Variables

The syntax for declaring reference variables is –

DATA <ref> TYPE REF TO <type> VALUE IS INITIAL.

System Variables

- ABAP system variables are accessible from all ABAP programs.
- These fields are actually filled by the run-time environment.
- The values in these fields indicate the state of the system at any given point of time.
- You can find the complete list of system variables in the SYST table in SAP

4.3 Processing Data

Arithmetic Calculations

Arithmetic Expressions

Arithmetic expressions are ABAP expressions with a combination of values, operators, and functions that the runtime system processes to calculate a result. For arithmetic expressions the result type depends on the type of the operands used as input to the expression.

You can use an arithmetic expression in any reading operand position, for example, the right-hand side of a value assignment.

total = amount1 + amount2
average = (2 * amount1 + 1 * amount2) / 5

Some Operators

- +
- (Addition)
- (Subtraction)
- *
- / (Multiplication)
- % (Division)
- MOD

Some Numeric Functions

- sqrt() (square root)
- int() (integer representation)

Sequence of Execution

- Functions before multiplication/division
- Multiplication/division before addition/subtraction
- From left to right if identical precedence

Note:

- ABAP syntax requires at least one blank between operators and operands.
- 1+1 is correct. 1+1 leads to a syntax error.
- Blanks are needed after opening brackets and before closing brackets.

Processing Strings

String templates are ABAP expressions of result type string. You can use string templates in any reading operand position, for example, the right-hand side of a value assignment.

SAP ABAP → String Template

Simple String Template: {<literal_text>} → String template containing only literal text

String Literal with Embedded Expression: {<text>}{<expression>}{<text>} → String template with Text and embedded expression

Examples:

{Hello World}	"literal text only"
{(amount1 + amount2)}	"1 Embedded expression"
{Total} { amount1 + amount2 } {sub}	"Text and 1 expression"

4.4 Internal Tables

Defining a Simple Internal Table

Internal Tables

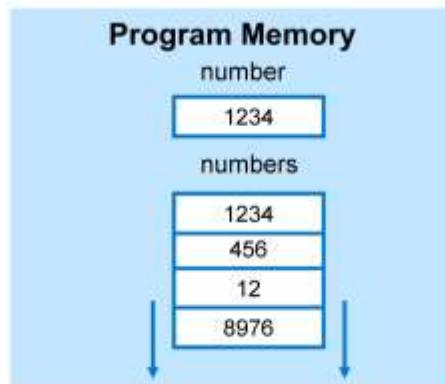
Internal tables are variable data objects in which you can store several values of identical type. This type has to be specified in the declaration and is called the row type of the internal table.

TYPE TABLE OF for declaring an Internal Table

```
DATA <internal_table> TYPE TABLE OF <row_type>.
```

ABAP Code Example

```
DATA number    TYPE i VALUE 1234.  
DATA numbers   TYPE TABLE OF i.  
  
APPEND 1234 TO numbers.  
APPEND 456   TO numbers.  
APPEND 12    TO numbers.  
APPEND 8976 TO numbers.  
...  
...
```



Each value occupies one row of the internal table. The number of rows is not restricted. Theoretically, you can store any number of values in one internal table. Limitations only come from technical boundaries like available memory or system configuration.

The initial value of an internal table is an empty table or, in other words, a table with 0 lines. There are different techniques for filling an internal table. The example uses the APPEND statement to add a new row at the end of the internal table and fill it with a value.

Table Types

The type of an internal table is called a table type.

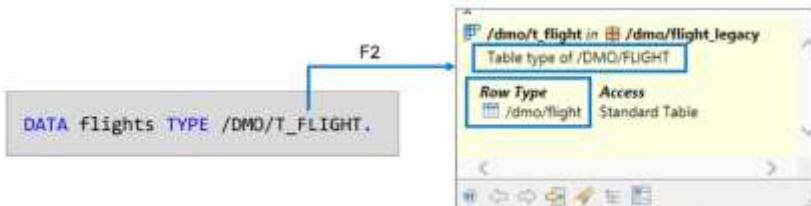
In the previous example we used TYPE TABLE OF in the DATA statement directly. The table type was bound to the declared variable.

As an alternative you can use TYPE TABLE OF in a TYPES statement to define a table type with a name. You can then use this table type, for example, in a DATA statement. The visibility of these types depends on the position of the TYPES statement.

Table Type with Statement TYPES

```
TYPES <table_type_name> TYPE TABLE OF <row_type>.  
DATA <internal_table> TYPE <table_type_name>.
```

Table Type in the ABAP Dictionary



Processing Data with a Simple Internal Table

Let's look at how you can process data with a simple internal table

Working with a Simple Internal Table

Select each step to learn more:

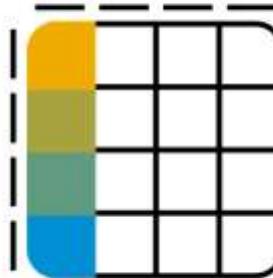
[Filling an Internal Table with APPEND](#)

[Emptying an Internal Table](#)

[Retrieve a Single Row from an Internal Table](#)

[Reading from an Internal Table in a Loop](#)

[Inline Declaration of Work Area](#)



4.5 Defining a Local Class

Object

An object is a special kind of variable that has distinct characteristics and behaviors. The characteristics or attributes of an object are used to describe the state of an object, and behaviors or methods represent the actions performed by an object.

An object is a pattern or instance of a class. It represents a real-world entity such as a person or a programming entity like variables and constants.

An object has the following three main characteristics –

- Has a state.
- Has a unique identity.
- May or may not display the behavior.

The state of an object can be described as a set of attributes and their values. For example, a bank account has a set of attributes such as Account Number, Name, Account Type, Balance, and values of all these attributes.

Objects can interact with one another by sending messages. Objects contain data and code to manipulate the data. An object can also be used as a user-defined data type with the help of a class. Objects are also called variables of the type class.

Class

A class is used to specify the form of an object and it combines data representation and methods for manipulating that data into one neat package. The data and functions within a class are called members of the class.

Class Definition and Implementation

When you define a class, you define a blueprint for a data type. This doesn't actually define any data, but it does define what the class name means, what an object of the class will consist of, and what operations can be performed on such an object. That is, it defines the abstract characteristics of an object, such as attributes, fields, and properties.

The following syntax shows how to define a class –

```
CLASS <class_name> DEFINITION.
```

.....

.....

```
ENDCLASS.
```

4.6 Create Instance of a Class

- Instances of a class can be created only where permitted by the addition CREATE of the statement CLASS DEFINITION.
- If the reference variable oref specified after CREATE OBJECT is passed simultaneously to the instance constructor, it points to the new object when this constructor is executed. To pass a reference to an existing object to the instance constructor, a different reference variable needs to be used.
- The statement CREATE OBJECT creates a heap reference. All references that point to the object or its parts are also heap references and keep the object alive. The same applies to field symbols that point to instance attributes or to their parts.
- When a class is used, the instance operator NEW acts like the statement CREATE OBJECT oref TYPE class and can be used in general expression positions.

Creating an Object

The object creation usually includes the following steps –

- Creating a reference variable with reference to the class. The syntax for which is

```
DATA: <object_name> TYPE REF TO <class_name>.
```

- Creating an object from the reference variable. The syntax for which is –

CREATE Object: <object_name>.

4.7 Defining Methods

Methods of Classes

Methods are internal procedures of a class that determine the behavior of an object. They can access all the attributes of all instances of their class and can therefore change the status of an object. Methods have a parameter interface, used by the system to pass values to them when they are called, and by which they can return values to the caller. The private attributes of a class can only be changed using methods of the same class.

Definition

A method *meth* is declared in the declaration part of a class using the statements METHODS and CLASS-METHODS and implemented in the implementation part of the class using the processing block

METHOD meth.

...

ENDMETHOD.

.As in all procedures, local data types and data objects can be declared in methods. Methods are called statically using the expression *meth(...)* or dynamically using the statement CALL METHOD (Dynamic Invoke).

Different Types of Methods

Instance Methods

Instance methods are declared using the METHODS statement. They can access all the attributes of a class and can trigger all its events.

Static Methods

Static methods are declared using the CLASS-METHODS statement. This statement can access static attributes of a class and can trigger static events only.

Constructors

As well as the normal methods that are called explicitly, there are two special methods called constructor and class_constructor, which are called automatically when an object is created or when a class component is accessed for the first time.

Functional Methods

Functional methods are methods with precisely one RETURNING parameter and any number of other formal parameters. Functional methods cannot just be called as

standalone statements, but also as functional method calls in operand positions for functions and expressions. Here they can be also be combined as method chainings.

Optional Methods

In interfaces, methods can be made optional using the addition DEFAULT of the statements METHODS and CLASS-METHODS. An optional interface method does not need to be implemented explicitly in a class when an interface is implemented. Instead, a default behavior is specified for calls of non-implemented methods in the definition. DEFAULT IGNORE calls an empty method and DEFAULT FAIL raises an exception.

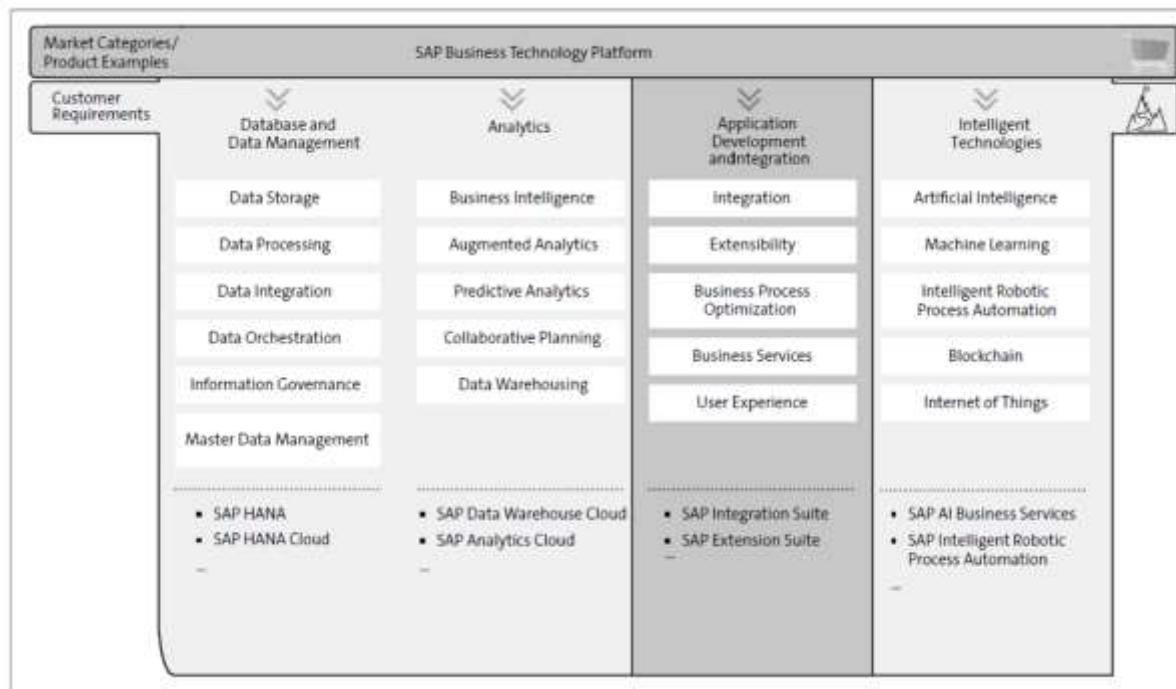
Unit 5: ABAP on SAP Business Technology Platform

Learning Outcomes:

- Introduction to SAP BTP
- Understanding role of ABAP on SAP BTP
- Checking the scenarios of ABAP on BTP

5.1 Introduction to SAP Business Technology Platform

SAP Business Technology Platform (SAP BTP) brings together data management, analytics, artificial intelligence, application development, automation, and integration in one, unified environment.



Personalize experiences for SAP applications

- Deliver innovations that natively integrate with SAP applications
- Enrich user interactions with artificial intelligence and automation
- Access real-time, complete views of all your data

Innovate faster with business context

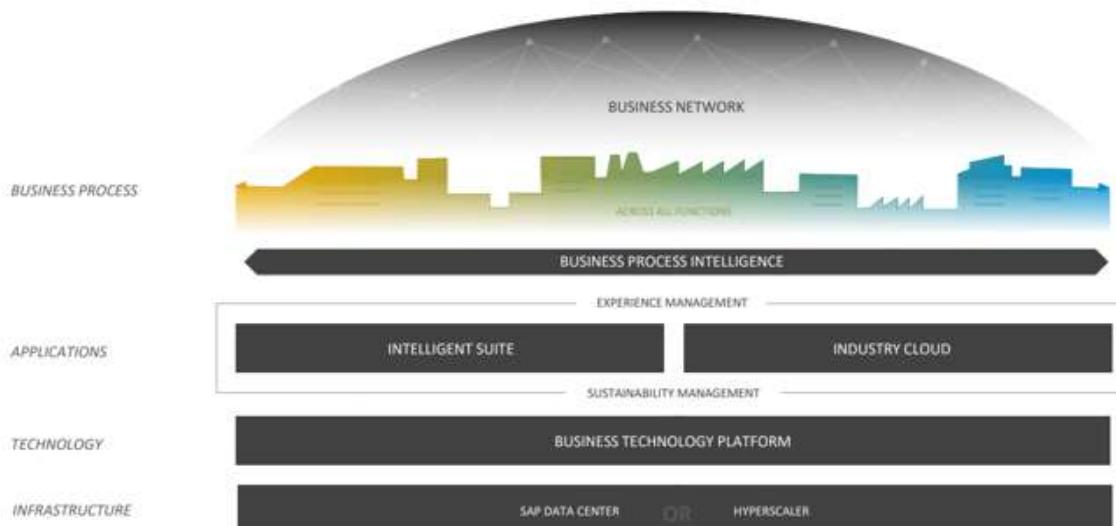
- Work faster with no-code and code-first development tools
- Use and analyze data from SAP applications within the right context and meaning
- Jump-start projects with prebuilt industry content and use cases

Run on a trusted, enterprise-grade platform

- Deploy in a mission-critical cloud environment managed by SAP
- Customize business processes without the need for maintenance
- Use your preferred cloud while interoperating with your existing IT landscape

INTELLIGENT ENTERPRISE

Evolved Vision



When to use what?



SAP BTP is designed for business transformation and not just technology transformation or optimization. The application development and integration pillar gives you

everything you need for agile business process innovation, extension, and integration in the cloud and across hybrid scenarios.

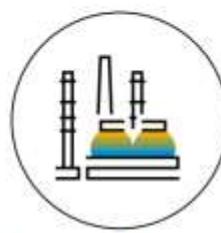
5.2 ABAP on SAP Business Technology Platform

SAP BTP ABAP Environment is the SAP Platform-as-a-Service (PaaS) offering for ABAP development that enables developers to leverage their traditional on-premise ABAP know-how to develop and run ABAP applications in the SAP Business Technology Platform, either as extension to SAP software or as standalone applications.

Motivation



Customer Base



Enterprise Readiness



Transition to Cloud

Huge **customer and partner base** running ABAP based solutions with custom code

ABAP as proven environment for **enterprise ready** business applications with competitive cost of development

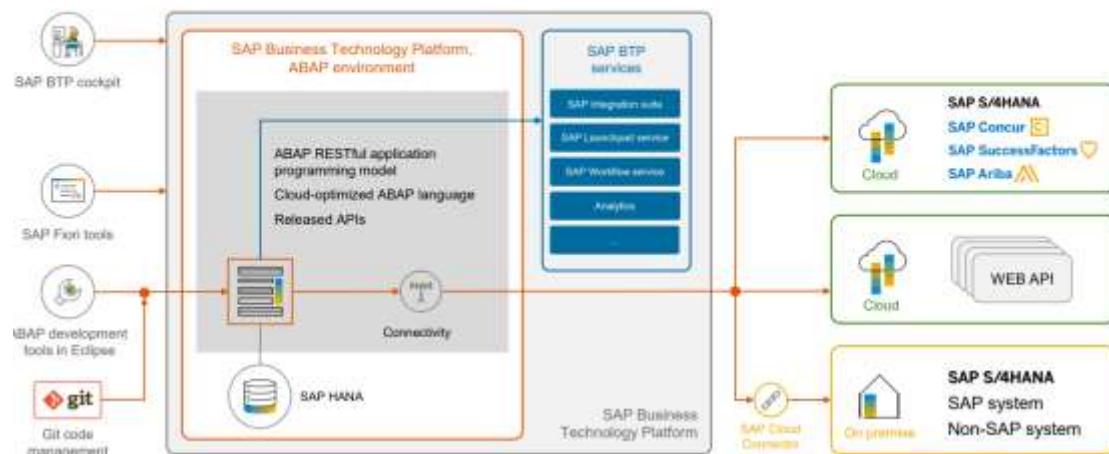
Growing market adoption of **SAP S/4HANA Cloud** with tailored extensibility for customers and partners

Value of SAP BTP ABAP Environment

- ▶ **Cloud ERP**
Transform your existing ABAP assets **to the cloud**
- ▶ **Innovation Platform**
Develop and run innovative ABAP apps **on a PaaS in the Cloud**
 - ▶ use always the **newest version of the ABAP Platform**
 - ▶ use always the **newest version of SAP HANA**
 - ▶ use always the **newest SAP BTP services**
 - ▶ **Delegate the operation** of the ABAP PaaS to SAP
- ▶ **Clean Core**
Decouple from your **core business system**
 - ▶ Delegate cloud integration scenarios to a cloud hub
 - ▶ Reduce upgrade costs with decoupled cloud extensions
 - ▶ Delegate non ERP user roles like consumer to a cloud solution



Vital parts of SAP Business Technology Platform (BTP), ABAP environment



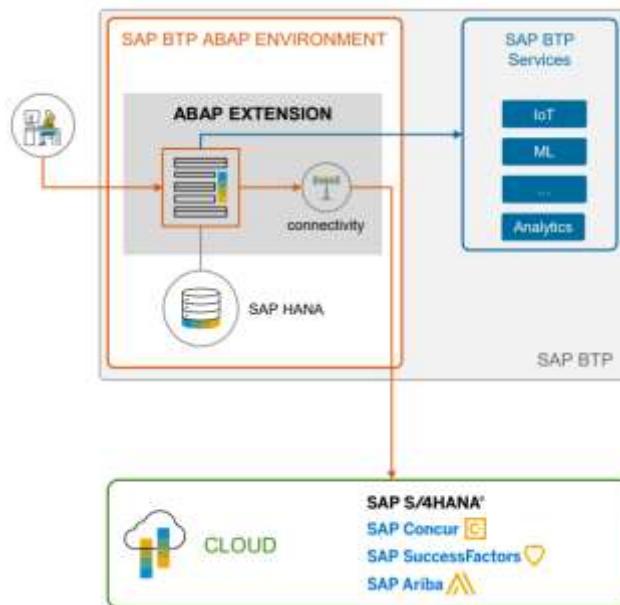
5.3 Scenarios of ABAP on SAP BTP

Extension scenario 1: Cloud ERP

Extend SAP S/4HANA Cloud or other SAP cloud offerings with cloud extensions

Use SAP BTP ABAP Environment to extend SAP S/4HANA Cloud or other

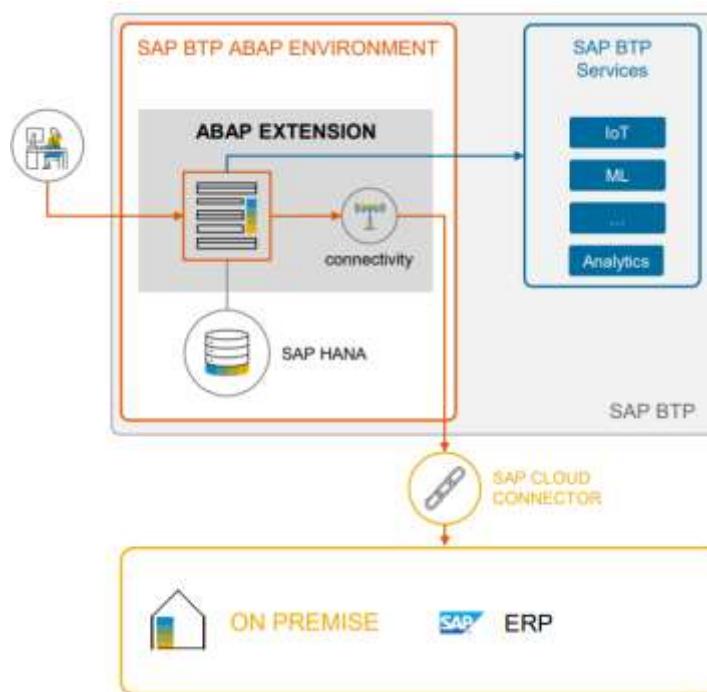
- SAP cloud solutions SAP Cloud solutions like SAP S/4HANA Cloud provide in-app extensibility to extend SAP apps and processes, but there is no support for classic custom ABAP development on top of SAP S/4HANA Cloud
- SAP BTP is the foundation to develop and run custom cloud extensions and the ABAP environment shall be used for ABAP based cloud extensions



Extension scenario 2: Innovation Platform

Develop and run innovative ABAP apps on a PaaS in the Cloud

- Benefit from the newest ABAP Platform and SAP HANA database technologies independent from your existing on-premise system landscape
- Build your Fiori apps with the new future proof ABAP RESTful Application Programming Model
- Utilize SAP BTP services like IoT, machine learning etc. in your cloud extension
- Delegate operation of the ABAP PaaS and new technologies to SAP



Extension scenario 3: Hub-like usage

Decouple ABAP implementations from your core business systems

EXTERNAL USER GROUP

Make your cloud app available to a broader audience that does not have access to your core business systems (e.g. consumer apps)

INTEGRATION HUB

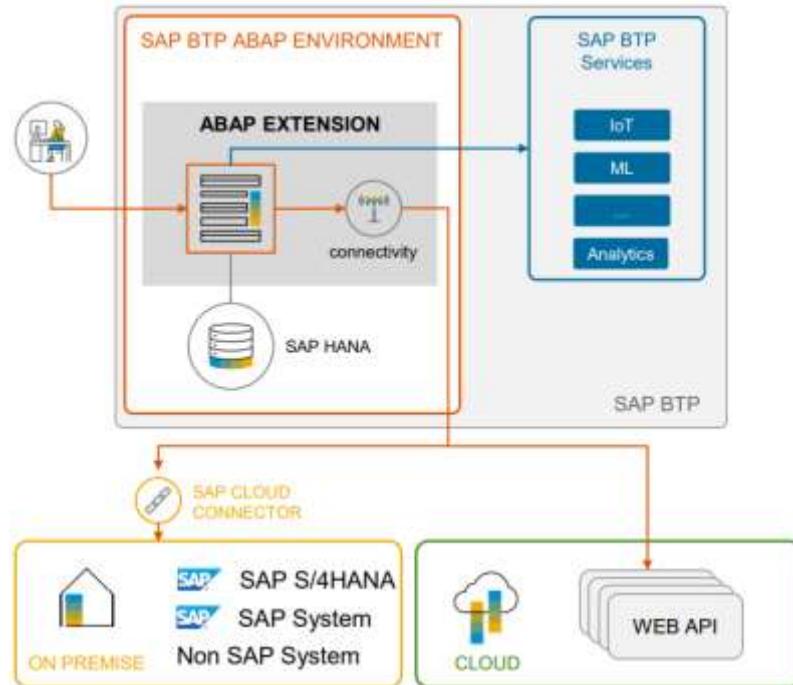
Integrate in your cloud extension multiple cloud/on-premise systems with SAP/non-SAP cloud services

DATA INTEGRATION

Collect data from multiple sources in your cloud extension for further processing and analysis

DECOUPLED EXTENSION

Cloud extensions use only well defined (remote) APIs of the Business system. This reduces the risk and effort for business system upgrades.



Unit 6: Hands-On SAP Business Technology Platform ABAP Environment

Learning Outcomes:

- Creating a Free trial of SAP BTP ABAP Environment
- To understand creating a Package, Class for SAP ABAP
- To perform some query operations on the Table

6.1 Practical: Creating a BTP ABAP Environment

Create an SAP BTP ABAP Environment Trial User

Create a trial user and ABAP cloud project with SAP BTP ABAP environment.

We will learn

- How to create a trial user
- How to create an ABAP Cloud project

Step 1: In your web browser, open the <https://cockpit.hanatrial.ondemand.com/>.

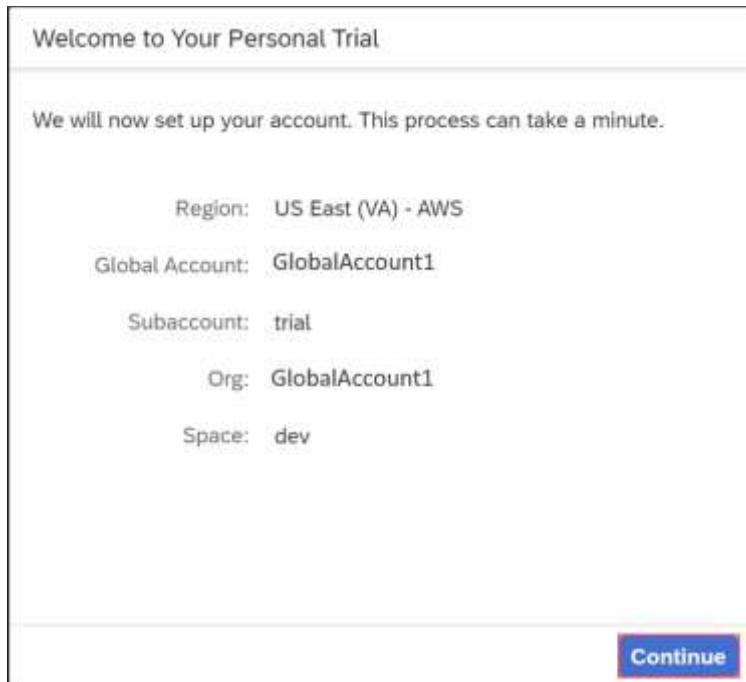
Navigate to the trial global account by clicking Go to Your Trial Account.

The screenshot shows the SAP BTP Cockpit interface. At the top, it says "Welcome to SAP BTP Trial". Below that is a button labeled "Go To Your Trial Account". The main area is titled "Quick Tool Access" and contains three items: "SAP Business Application Studio" (with a description about developing business applications using SAP's code generator, Web-based IDE), "CLI for BTP" (with a description about managing trial account using the command-line interface), and "APIs for SAP BTP" (with a description about managing, building, and extending the core capabilities of SAP BTP). At the bottom, there are two sections: "Start with Tutorials" containing links to "Build a Business Application Using CAP for Node.js" (Extension Suite - Development), "Get Started with SAP Mobile Cards" (Enterprise Suite - Digital Experience), and "Request Product Details with an Integration Scenario" (Integration Suite); and a "Footer" section with links to "SAP BTP Documentation", "SAP BTP Support", and "SAP BTP Community".

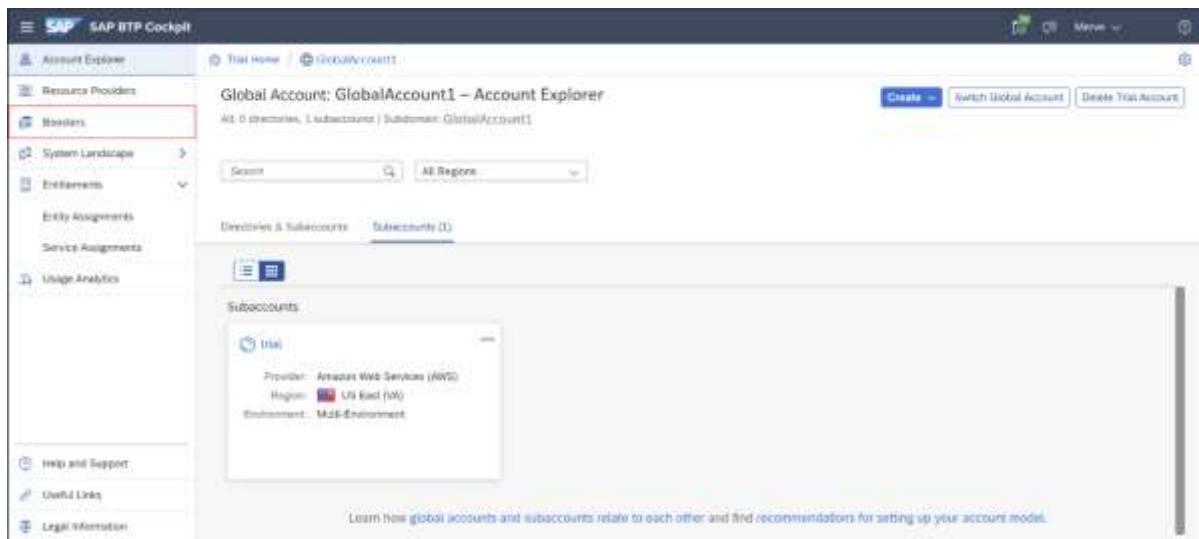
If this is your first time accessing your trial account, you'll have to configure your account by choosing a region. Please select US East (VA) as a region. Your user profile will be set up for you automatically.

Wait till your account is set up and ready to go. Your global account, your subaccount, your organization, and your space are launched. This may take a couple of minutes.

Choose Continue.



From your global account page, choose Boosters on the left side.



Search the Prepare an Account for ABAP Trial tile and press Start to start your booster. If you already created a service instance and service key, then please skip this step and move on with Step 2. Only one service instance can be created at a time.

The screenshot shows the SAP BTP Cockpit interface. On the left, there's a navigation sidebar with options like Account Explorer, Resource Providers, Boosters (which is selected), System Landscape, Entitlements, Entity Assignments, Service Assignments, and Usage Analytics. Below the sidebar, there's a search bar and a note about using guided boosters to build applications or use different platform services and features. The main area is titled 'Booster Suite - Development Efficiency (B)' and contains five cards:

- Prepare an Account for ABAP Trial**: Includes a 'Create instance and service key' step and a 'Start' button.
- Prepare an Account for SAP Customer Order Sourcing**: Includes an 'Assign entitlements' step and a 'Start' button.
- Set up account for Data Attribute Recommendation**: Includes a note about getting access to the Data Attribute Recommendation API and a 'Start' button.
- Set up account for Document Information Extraction**: Includes a 'Start' button.
- Set up account for Service Ticket Intelligence**: Includes a 'Start' button.

Now the service instance and service key will be created for the ABAP trial user. The service key can be found inside the service instance.

The screenshot shows a progress dialog box with the title 'Progress'. It lists two tasks:

- Assigning Service Quotas**: Status: **DONE**
- Creating Service Instances**: Status: **PROCESSING**

The booster is now executed successfully. Download your service key for later use.

The screenshot shows a success dialog box with the following content:

- Success** (indicated by a green checkmark icon)
- Booster executed successfully**
- Your subaccount is ready for ABAP Development.**
- [Navigate to Subaccount](#)
- [Go to Instance](#)
- [Go to Service Key](#)
- [Download Service Key](#) (The download icon is highlighted with a red box)
- [Close](#)

Step 2: Open ABAP Development Tools

Open Eclipse. Make sure you have installed ADT in your Eclipse.

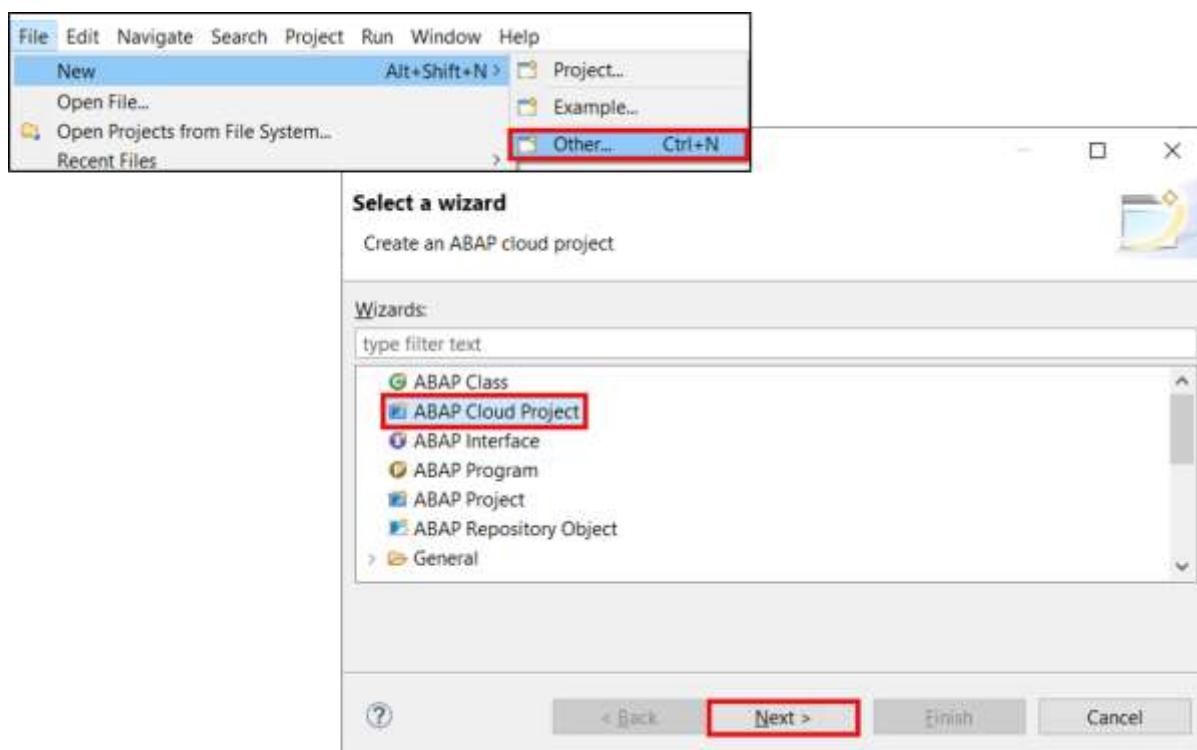
First, we need to install Eclipse IDE to perform the practical's. We need to follow the following steps to configure Eclipse IDE on local system.

To install the front-end component of ADT, proceed as follows:

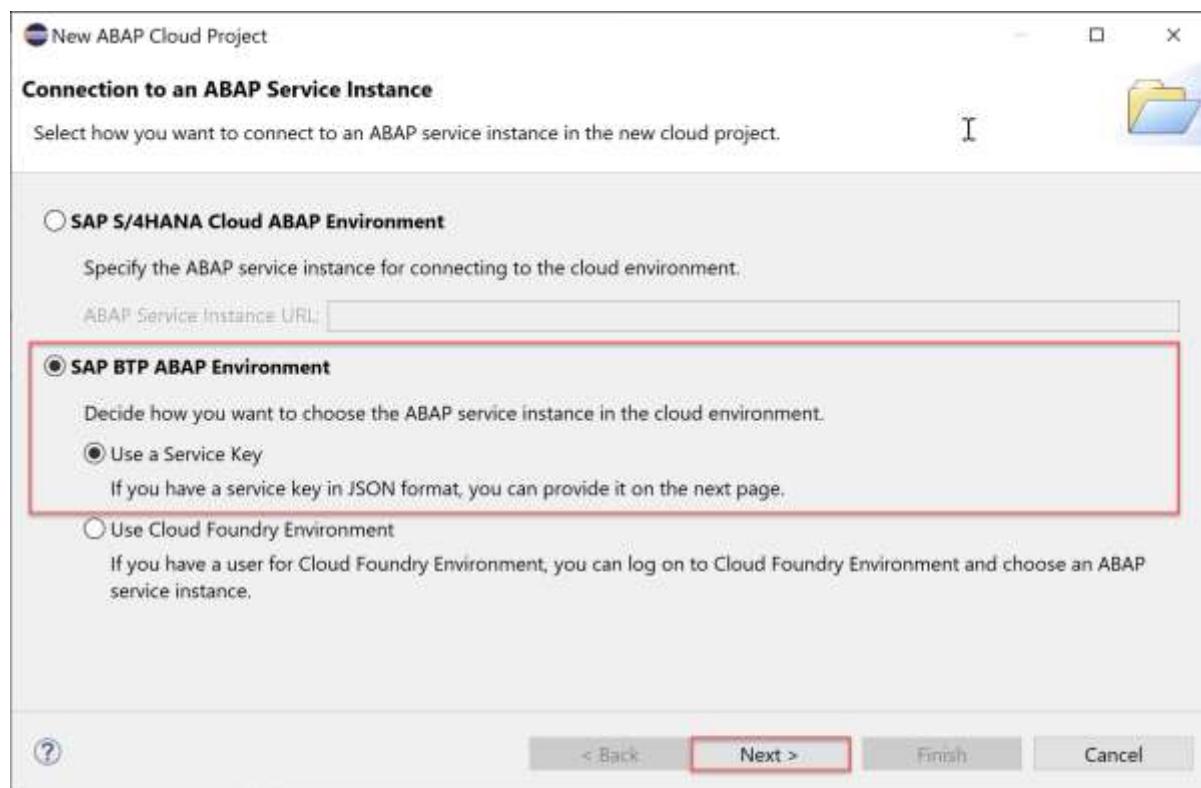
1. Get an installation of Eclipse 2022-09 (x86_64) ([Eclipse IDE for Java Developers](#))
2. In Eclipse, choose in the menu bar Help > Install New Software...
3. Enter the URL <https://tools.hana.ondemand.com/latest>
4. Press Enter to display the available features.
5. Select ABAP Development Tools and choose next.
6. On the next wizard page, you get an overview of the features to be installed. Choose Next.
7. Confirm the license agreements and choose Finish to start the installation.

Step 3: Create ABAP cloud project

1. Select File > New > Other > ABAP Cloud Project and click Next >.



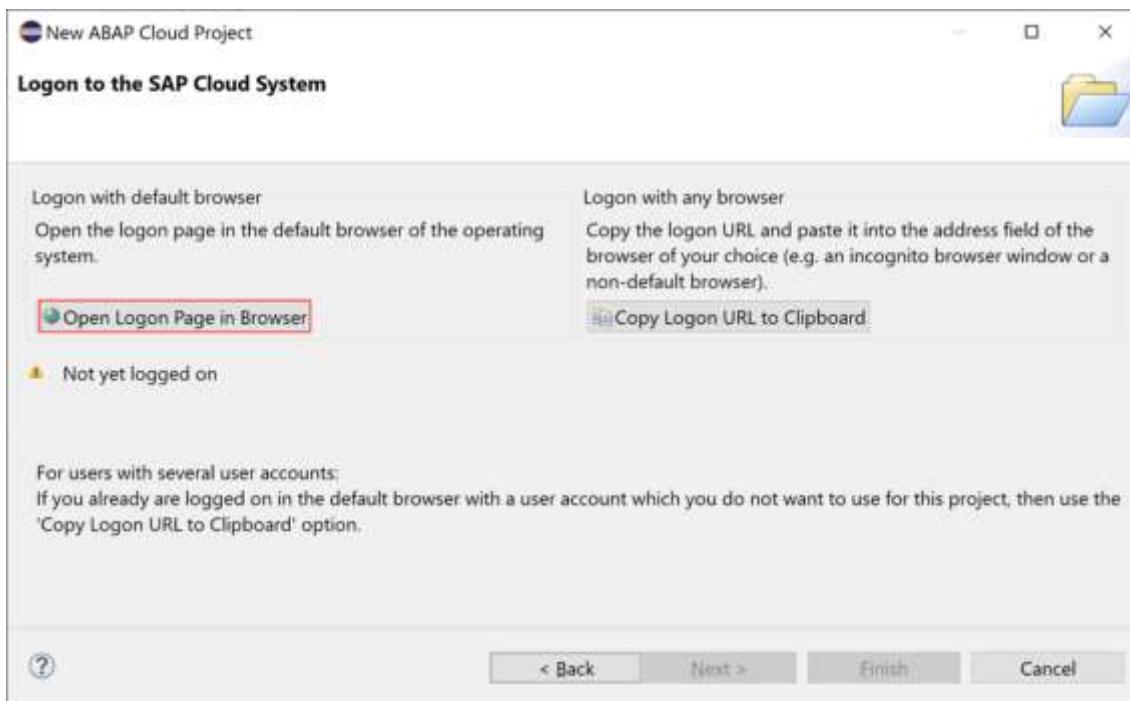
2. Select SAP BTP ABAP Environment > Use a Service Key and click Next >.



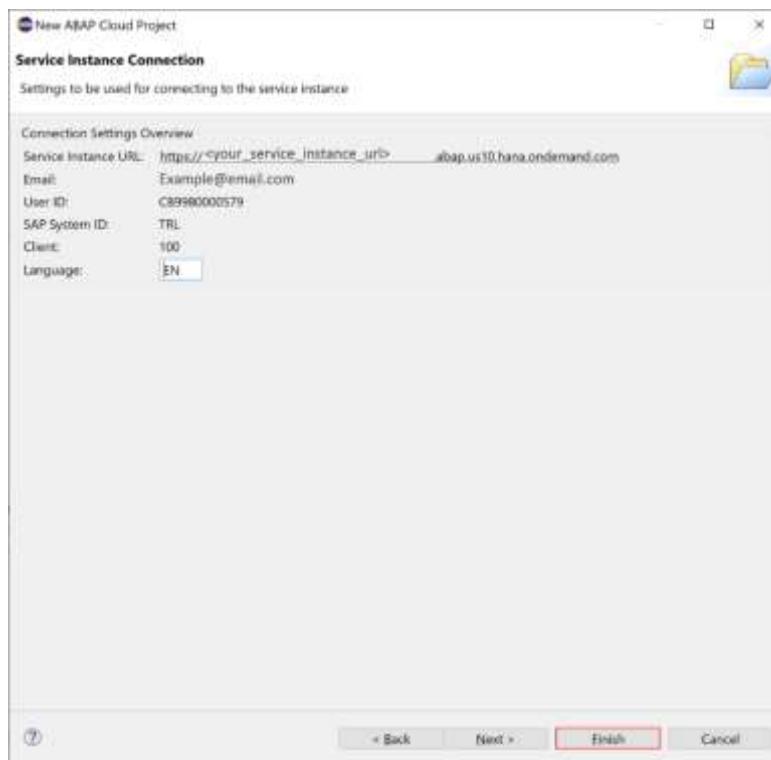
3. Import your service key and click Next >.



4. Click Open Logon Page in Browser.

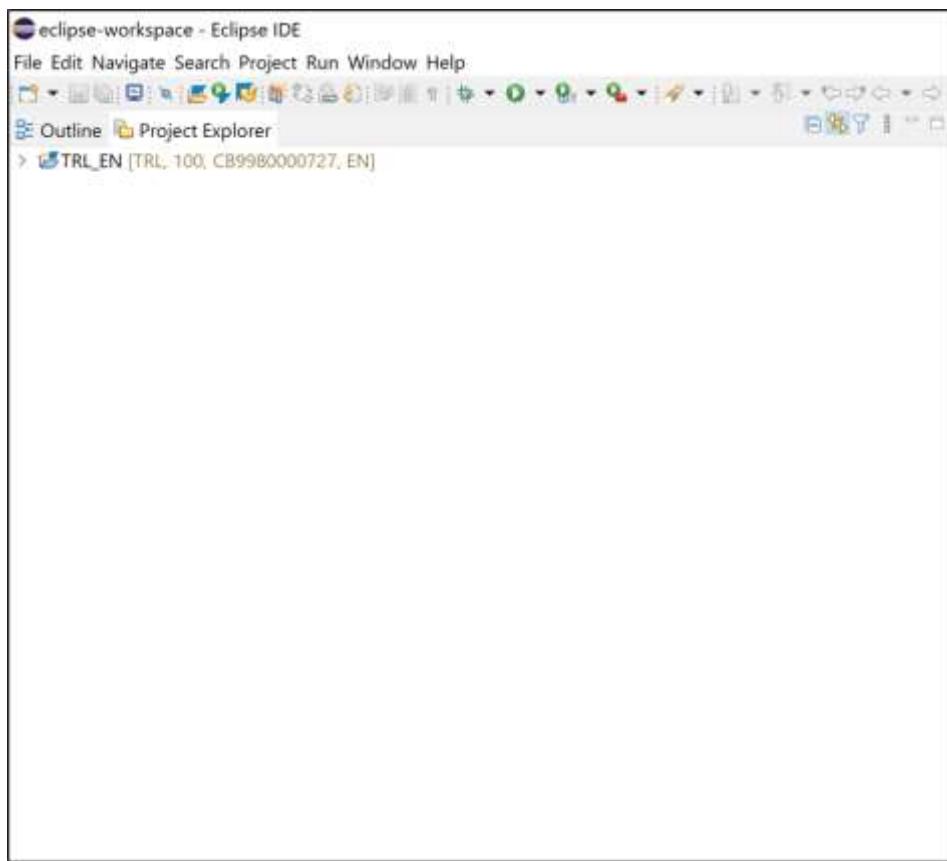


5. Now you've been authenticated automatically. Provide your credentials if requested. The credentials are the same you used to create your trial account on SAP BTP.



Click Finish.

6. Your trial system appears on the project explorer.

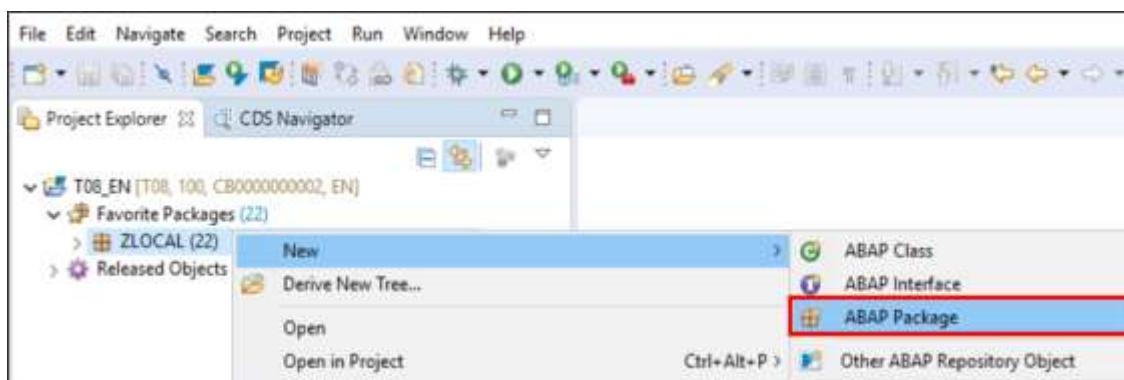


6.2 Practical: Creating an ABAP Package

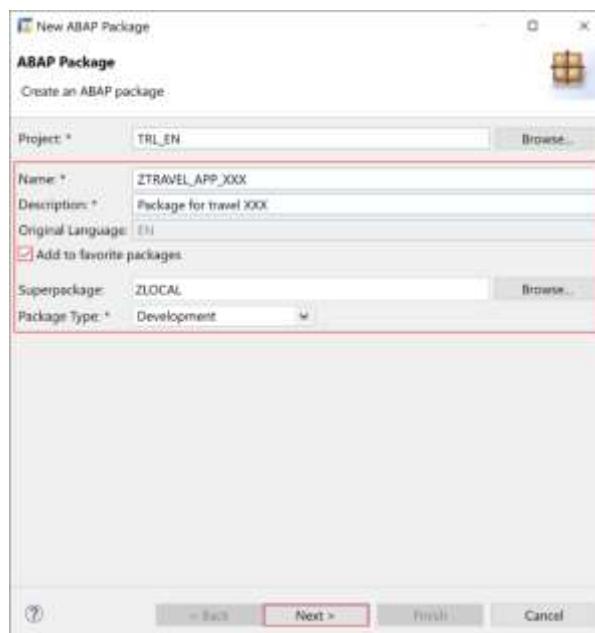
In this practical, wherever XXX appears, use a number.

1. Open ABAP Development Tools (ADT) and select your ABAP Cloud Project you created in Create a SAP BTP ABAP Environment Trial User.

Right-click on ZLOCAL and select New > ABAP Package.

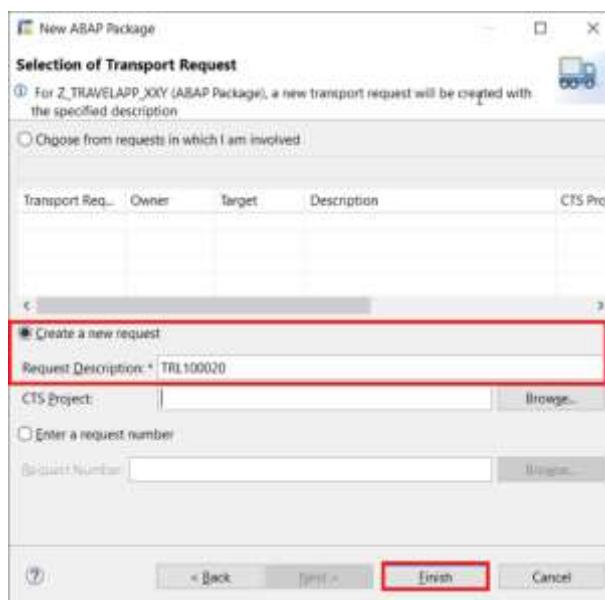


2. Create a new ABAP package:
 - Name: ZTRAVEL_APP_XXX
 - Description: Package for travel XXX
 - Superpackage: ZLOCAL
 - Check Add to favorite packages.



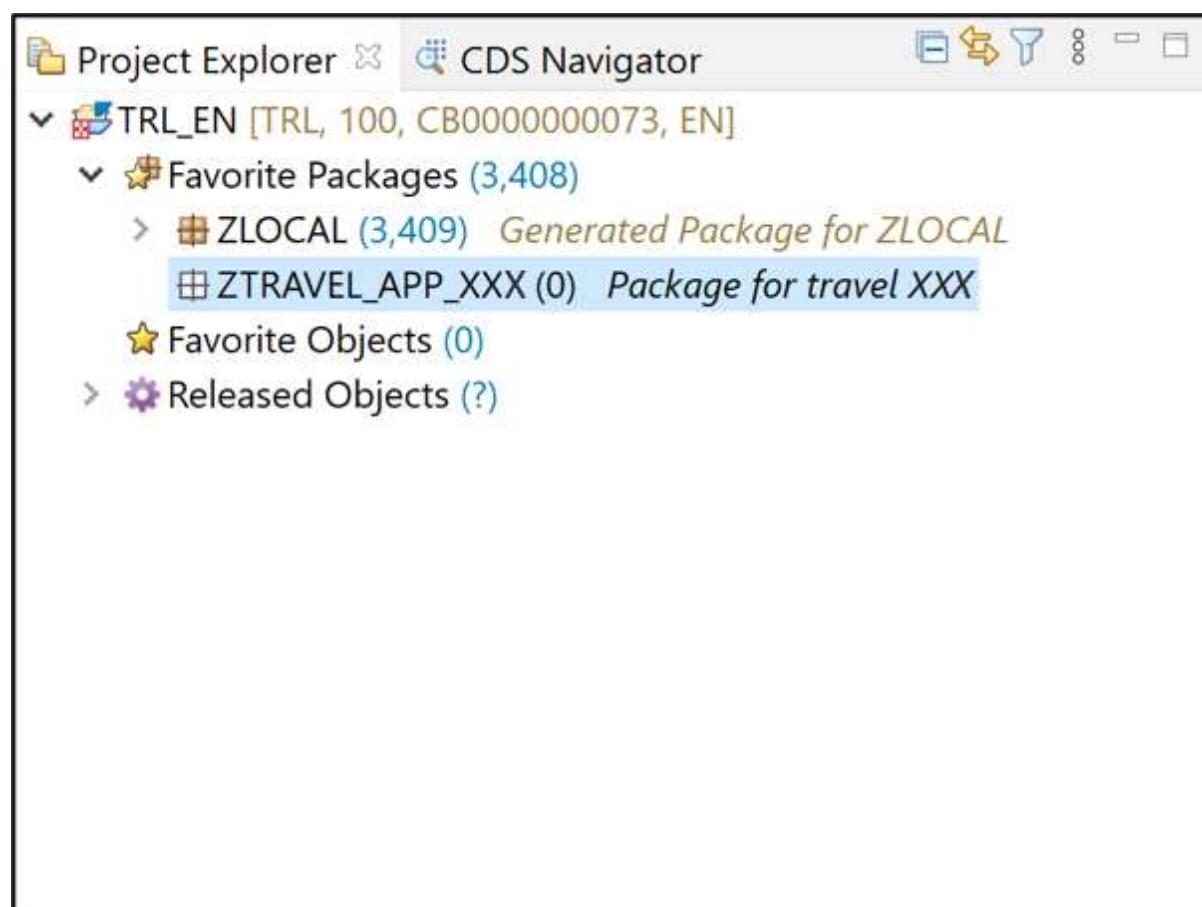
Click Next >.

3. Select Create new request and enter a request description.



Click Finish.

4. Now your package is added to favorite objects.

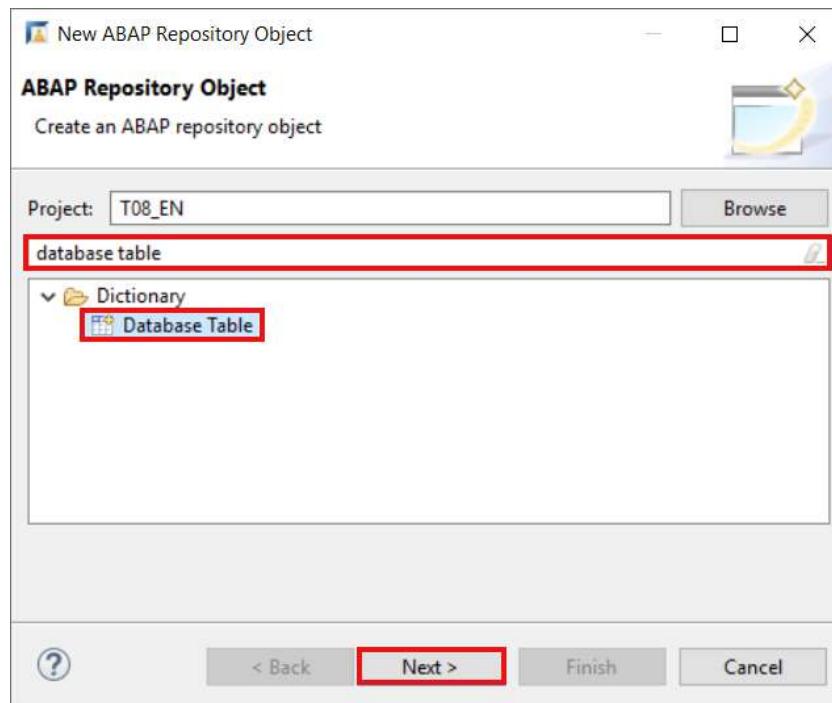


6.3 Practical: Creating a Database Table

1. Right-click on your package ZTRAVEL_APP_XXX, select New > Other ABAP Repository Object.



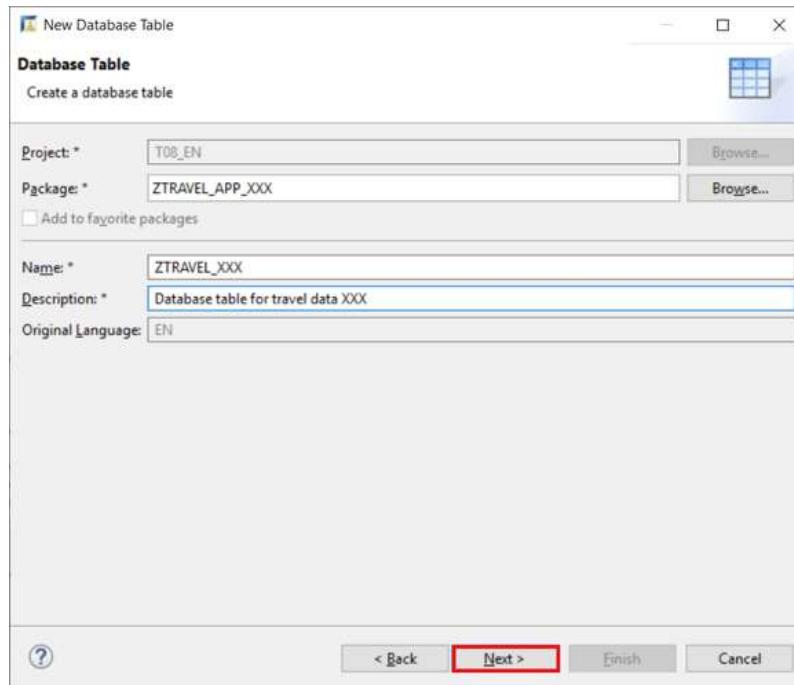
2. Search for database table, select it and click Next >.



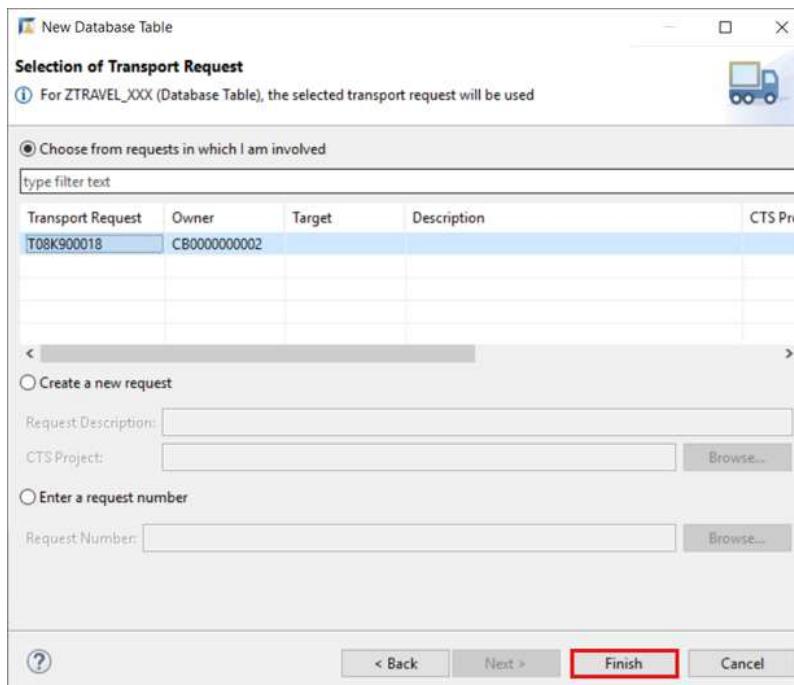
3. Create a new database table:

- Name: ZTRAVEL_XXX
- Description: Database table for travel data XXX

Click Next >.



4. Click Finish to create your transport request.



5. Replace your code with following:

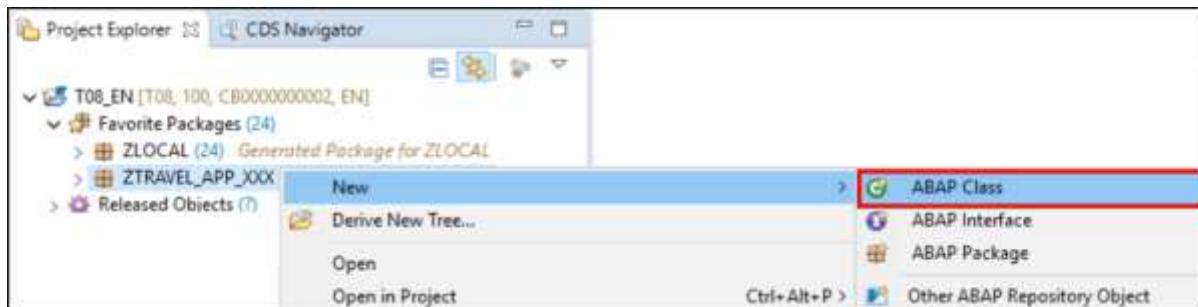
```
@EndUserText.label : 'Database table for travel data XXX'  
@AbapCatalog.enhancementCategory : #NOT_EXTENSIBLE  
@AbapCatalog.tableCategory : #TRANSPARENT  
@AbapCatalog.deliveryClass : #A  
@AbapCatalog.dataMaintenance : #RESTRICTED  
define table ztravel_xxx {  
    key client      : abap.clnt not null;  
    key mykey       : sysuuid_x16 not null;  
    travel_id      : /dmo/travel_id;  
    agency_id      : /dmo/agency_id;  
    customer_id    : /dmo/customer_id;  
    begin_date     : /dmo/begin_date;  
    end_date       : /dmo/end_date;  
    @Semantics.amount.currencyCode : 'ztravel_xxx.currency_code'  
    booking_fee    : /dmo/booking_fee;  
    @Semantics.amount.currencyCode : 'ztravel_xxx.currency_code'  
    total_price    : /dmo/total_price;  
    currency_code  : /dmo/currency_code;  
    description    : /dmo/description;  
    overall_status : /dmo/overall_status;  
    created_by     : syuname;  
    created_at     : timestamppl;  
    last_changed_by : syuname;  
    last_changed_at : timestamppl;  
}
```

6. Save and Activate

- Save – Ctrl + S
- Activate – Ctrl+F3

6.4 Practical: Create an ABAP Class

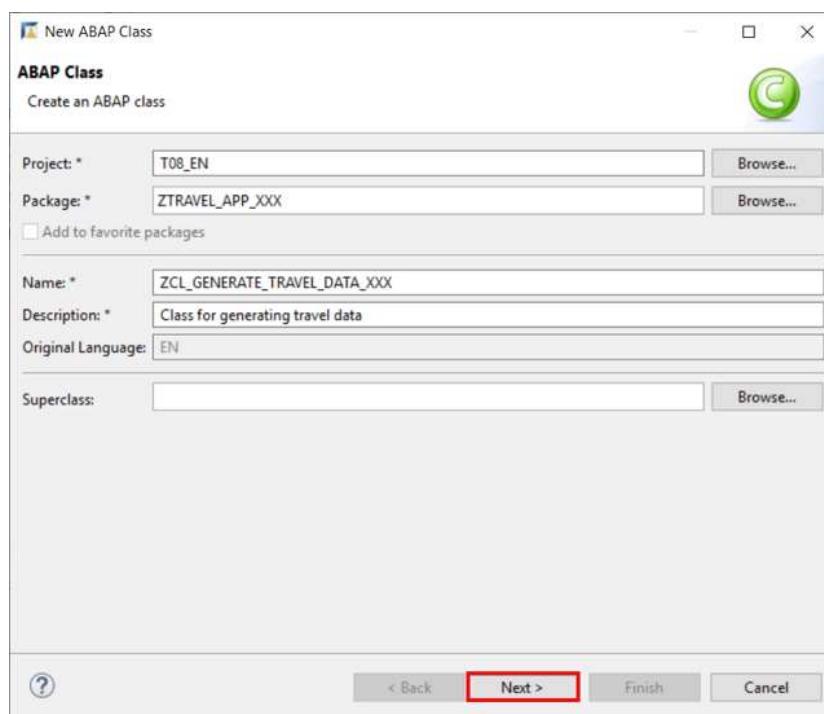
1. Right-click on your package ZTRAVEL_APP_XXX, select New > ABAP Class.



2. Create a new ABAP class:

- Name: ZCL_GENERATE_TRAVEL_DATA_XXX
- Description: Class for generating travel data

Click Next >.



3. Click Finish to create your transport request.

New ABAP Class

Selection of Transport Request

(i) For ZCL_GENERATE_TRAVEL_DATA_XXX (Global Class), the selected transport request will be used

Choose from requests in which I am involved

type filter text

Transport Request	Owner	Target	Description	CTS Pro
T08K900018	CB0000000002			

Create a new request

Request Description:

CTS Project:

Enter a request number

Request Number:

6.5 Practical: WAP to print Hello World

Create a New Class “zcl_generate_travel_data_yyy” and paste the following code.

```
CLASS zcl_generate_travel_data_yyy DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    INTERFACES if_oo_adt_classrun.
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.
```

```
CLASS zcl_generate_travel_data_yyy IMPLEMENTATION.
  METHOD if_oo_adt_classrun~main.
```

* output the result as a console message.

```
  out->write( '|Hello World |').
ENDMETHOD.
```

```
ENDCLASS.
```

Output:



6.6 Practical: Perform Query Operation on the Output Table

1. How to extract the records having Total Price > 700

Click on “Add Filter”. Click on Total Price and write the condition

The screenshot shows the SAP BusinessObjects Data Services Designer interface. In the center, there is a "Data Preview" window displaying a table of travel data. The columns include AGENCY_ID, CUSTOMER_ID, BOOKING_DATE, END_DATE, BOOKING_ID, TOTAL_PRICE, CURRENCY_CODE, and DESCRIPT. There are four rows of data. On the right side of the preview window, there is a "Filters" section. Under the "TOTAL_PRICE" filter, there is a dropdown menu with the condition " > 700 ".

AGENCY_ID	CUSTOMER_ID	BOOKING_DATE	END_DATE	BOOKING_ID	TOTAL_PRICE	CURRENCY_CODE	DESCRIPT
110001	00000177	2019-06-24	2019-06-28	600001	60.00	USD	one way
110001	000011	2019-06-10	2019-07-14	600002	17.00	AFN	Enter your code
110001	0000377	2019-06-24	2019-06-28	600003	60.00	USD	one way
110001	0000115	2019-06-10	2019-07-14	600004	17.00	USD	Enter your code

2. Click on Select Columns and Select the columns you want to display

The screenshot shows the SAP BusinessObjects Data Services Designer interface. In the center, there is a "Data Preview" window displaying a table of travel data. The columns shown are TRAVEL_ID, CURRENCY_CODE, and CLIENT. There are six rows of data.

TRAVEL_ID	CURRENCY_CODE	CLIENT
00000003	USD	100
00000100	AFN	100
00000101	AFN	100
00000009	USD	100
00000110	AFN	100
00001115	AFN	100

3. Filter the Country Code = 'AFN' from the data

The screenshot shows the SAP BusinessObjects Data Services Designer interface. On the left, the Project Explorer displays various objects like 'ZTRAVEL_APP_XX' and 'ZTRAVEL_APP_XA'. In the center, the 'Data Preview' window shows a table with four rows of travel data. The columns are: ISBN_DATE, INVOICE_DATE, TOTAL_PRICE, and CURRENCY_CODE. The last column has a dropdown filter set to 'AFN'. The preview indicates 4 rows retrieved in 25 ms.

ISBN_DATE	INVOICE_DATE	TOTAL_PRICE	CURRENCY_CODE
2019-06-11	2019-07-16	460.00	AFN
08001	2019-06-18	17.00	AFN
08002	2019-06-11	460.00	AFN
08011	2019-06-18	17.00	AFN

6.7 Practical: Perform various ABAP data types of operation

Numeric Operations:

Addition:

```
DATA lv_num1 TYPE i VALUE 10.
```

```
DATA lv_num2 TYPE i VALUE 5.
```

```
DATA lv_sum TYPE i.
```

```
lv_sum = lv_num1 + lv_num2.
```

```
WRITE: 'Sum:', lv_sum.
```

Subtraction:

```
DATA lv_num1 TYPE i VALUE 10.
```

```
DATA lv_num2 TYPE i VALUE 5.
```

```
DATA lv_diff TYPE i.
```

```
lv_diff = lv_num1 - lv_num2.
```

WRITE: 'Difference:', lv_diff.

Multiplication:

```
DATA lv_num1 TYPE i VALUE 10.  
DATA lv_num2 TYPE i VALUE 5.  
DATA lv_product TYPE i.
```

lv_product = lv_num1 * lv_num2.

WRITE: 'Product:', lv_product.

Division:

```
DATA lv_num1 TYPE f VALUE 10.0.  
DATA lv_num2 TYPE f VALUE 5.0.  
DATA lv_quotient TYPE f.
```

lv_quotient = lv_num1 / lv_num2.

WRITE: 'Quotient:', lv_quotient.

2-Character/String Operations:

Concatenation:

```
DATA lv_str1 TYPE string VALUE 'Hello'.  
DATA lv_str2 TYPE string VALUE 'World'.  
DATA lv_result TYPE string.
```

CONCATENATE lv_str1 lv_str2 INTO lv_result.

WRITE: 'Result:', lv_result.

String Length:

```
DATA lv_str TYPE string VALUE 'Hello World'.
```

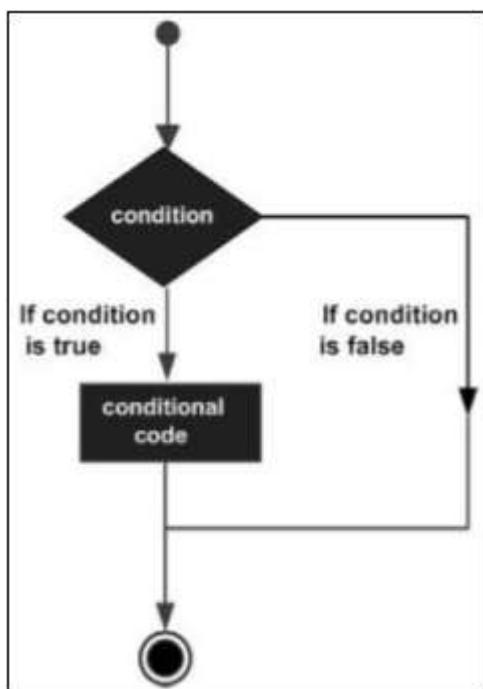
```
DATA lv_length TYPE i.
```

```
lv_length = STRLEN( lv_str ).
```

```
WRITE: 'Length:', lv_length.
```

ABAP programming language for decision-making statements.

the general form of a typical decision-making structure found in most of the programming languages –



S.No.	Statement & Description
1	<u>IF Statement</u> An IF statement consists of a logical expression followed by one or more statements.
2	<u>IF.. Else Statement</u>

	An IF statement can be followed by an optional ELSE statement that executes when the expression is false.
3	Nested IF Statement You may use one IF or ELSEIF statement inside another IF or ELSEIF statement.
4	CASE Control Statement CASE statement is used when we need to compare two or more fields or variables.

Example

```
DATA lv_age TYPE i.
```

```
lv_age = 25.
```

```
IF lv_age >= 18.
```

```
  WRITE: 'You are an adult.'.
```

```
ELSE.
```

```
  WRITE: 'You are not an adult.'.
```

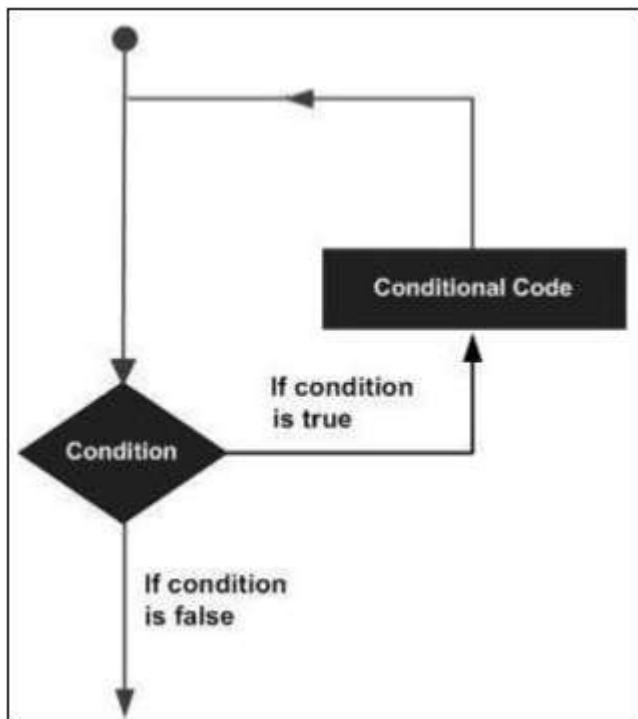
```
ENDIF.
```

In this example, we have a variable lv_age assigned a value of 25. The IF statement checks whether lv_age is greater than or equal to 18. If the condition is true, it executes the statements within the IF block and displays "You are an adult." Otherwise, it executes the statements within the ELSE block and displays "You are not an adult."

The output will be: ---- You are an adult.

6.8 Practical: Loops in ABAP, For loop, While Loop

Programming languages provide various control structures that allow for more complicated execution paths. A **loop statement** allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages.



Code For Loop Example:

```
DATA: lv_counter TYPE i.
```

```
lv_counter = 1.
```

```
DO 5 TIMES.
```

```
  WRITE: 'Counter:', lv_counter.
```

```
  lv_counter = lv_counter + 1.
```

```
ENDDO.
```

Explanation: In this example, we have a variable lv_counter initialized with a value of 1. The DO loop is executed 5 times. Within the loop, the current value of lv_counter is displayed using the WRITE statement. After each iteration, the value of lv_counter is incremented by 1 using the assignment statement lv_counter = lv_counter + 1.

The output will be:

Counter: 1

Counter: 2

Counter: 3

Counter: 4

Counter: 5

While Loop Example:

DATA: lv_counter TYPE i.

lv_counter = 0.

WHILE lv_counter < 5.

 WRITE: 'Counter:', lv_counter.

 lv_counter = lv_counter + 1.

ENDWHILE.

Explanation: In this example, the WHILE loop is executed as long as lv_counter is less than 5. The loop starts with lv_counter initialized to 0, and within the loop, the current value of lv_counter is displayed using the WRITE statement. After each iteration, the value of lv_counter is incremented by 1 using the assignment statement lv_counter = lv_counter + 1.

The output will be:

Counter: 0

Counter: 1

Counter: 2

Counter: 3

Counter: 4

6.9 Practical: Insert Data Table Entries and print them on the console

Open the class first created “zcl_generate_travel_data_xxx” and paste the following code.

```

CLASS zcl_generate_travel_data_xxx DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .
  PUBLIC SECTION.
    INTERFACES if_oo_adt_classrun.
  PROTECTED SECTION.
    PRIVATE SECTION.
ENDCLASS.

CLASS zcl_generate_travel_data_xxx IMPLEMENTATION.
  METHOD if_oo_adt_classrun~main.

    DATA itab TYPE TABLE OF ztravel_xxx.
*   fill internal travel table (itab)
    itab = VALUE #(
      ( mykey = '02D5290E594C1EDA93815057FD946624' travel_id = '00000022'
agency_id = '070001' customer_id = '000077' begin_date = '20190624' end_date
= '20190628' booking_fee = '60.00' total_price = '750.00' currency_code =
'USD'
          description = 'mv' overall_status = 'A' created_by = 'MUSTERMANN'
created_at = '20190612133945.5960060' last_changed_by = 'MUSTERFRAU'
last_changed_at = '20190702105400.3647680' )
      ( mykey = '02D5290E594C1EDA93815C50CD7AE62A' travel_id = '00000106'
agency_id = '070005' customer_id = '000005' begin_date = '20190613' end_date
= '20190716' booking_fee = '17.00' total_price = '650.00' currency_code =
'AFN'
          description = 'Enter your comments here' overall_status = 'A' cre-
ated_by = 'MUSTERMANN' created_at = '20190613111129.2391370' last_changed_by
= 'MUSTERMANN' last_changed_at = '2019071140753.1472620' )
      ( mykey = '02D5290E594C1EDA93858EED2DA2EB0B' travel_id = '00000103'
agency_id = '070010' customer_id = '000011' begin_date = '20190610' end_date
= '20190714' booking_fee = '17.00' total_price = '800.00' currency_code =
'AFN'
          description = 'Enter your comments here' overall_status = 'X' cre-
ated_by = 'MUSTERFRAU' created_at = '20190613105654.4296640' last_changed_by
= 'MUSTERFRAU' last_changed_at = '20190613111041.2251330' )
    ).

*   delete existing entries in the database table
    DELETE FROM ztravel_xxx.

*   insert the new table entries
    INSERT ztravel_xxx FROM TABLE @itab.

*   output the result as a console message
    out->write( |{ sy-dbcnt } travel entries inserted successfully!| ).
  ENDMETHOD.
ENDCLASS.

```

1. Save, activate and click F9 to run your ABAP class.

- Save – Ctrl + S
- Activate – Ctrl + F3
- Run – F9

Check your result. Therefore open your database table ZTRAVEL_XXX and press F8 to see your data. Now the dictionary tables are filled with data.

Raw Data										
Filter paths: 3 rows retrieved - 20 ms										
CLIENT	MYKEY	TRAVEL_ID	AGENCY_ID	CUSTOMER_ID	BEGIN_DATE	END_DATE	BOOKING_FEE	TOTAL_PRICE	CURRENCY_CODE	DESCRIPTION
100	02D5290...	00000022	070001	000077	2019-06-24	2019-06-28	60.00	750.00	USD	rrr
100	02D5290...	00000106	070005	000005	2019-06-13	2019-07-16	17.00	650.00	AFN	Enter your com.
100	02D5290...	00000103	070010	000011	2019-06-10	2019-07-14	17.00	800.00	AFN	Enter your com.

Reference

1. <https://www.javatpoint.com/logistic-regression-in-machine-learning>
2. <https://www.datasciencecentral.com/profiles/blogs/understanding-the-applications-of-probability-in-machine-learning>
3. <https://www.allerin.com/blog/how-to-fine-tune-your-artificial-intelligence-algorithms>
4. <https://www.mygreatlearning.com/blog/gridsearchcv/>
5. <https://www.kdnuggets.com/2020/05/hyperparameter-optimization-machine-learning-models.html>
6. <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>
7. <https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589>
8. <https://medium.com/analytics-vidhya/role-of-distance-metrics-in-machine-learning-e43391a6bf2e>
9. <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
10. <https://towardsdatascience.com/basic-probability-theory-and-statistics-3105ab637213>
11. <https://www.javatpoint.com/machine-learning-naive-bayes-classifier>
12. [Andreas C. Müller and Sarah Guido , Introduction to Machine learning with Python , O'reilly , October 2016.](#)
13. <https://www.guru99.com/unsupervised-machine-learning.html#:~:text=Unsupervised%20Learning%20is%20a%20machine,deals%20with%20the%20unlabelled%20data.>
14. <https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeeccb78b422a>
15. <https://www.geeksforgeeks.org/clustering-in-machine-learning/#:~:text=Clustering%20is%20the%20task%20of,data%20points%20in%20other%20groups.>
16. <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>
17. [Chire, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons\](#)
18. <https://www.broadbandsearch.net/blog/internet-statistics>
19. <https://www.cloudcredential.org/blog/knowledge-byte-building-blocks-of-iot-architecture/>
20. <https://www.engineersgarage.com/iot-building-blocks-and-architecture-iot-part-2/>
21. <https://www.c-sharpcorner.com/UploadFile/f88748/internet-of-things-part-2/>
22. <https://www.oracle.com/in/internet-of-things/what-is-iot/>
23. <https://www.simplilearn.com/iot-applications-article>
24. <https://electronicscoach.com/electronic-components.html>

25. https://en.wikipedia.org/wiki/Electronic_component#:~:text=An%20electronic%20component%20is%20any,electrons%20or%20their%20associated%20fields.
26. <https://tesckt.com/transistor-application-circuits-and-it-application-in-daily-life/>
27. <https://www.elprocus.com/buzzer-working-applications/>
28. <https://www.vedantu.com/iit-jee/basic-logic-gates>
29. <https://www.monolithicpower.com/en/analog-vs-digital-signal>
30. <https://byjus.com/physics/pulse-width-modulation/>
31. <https://dengarden.com/home-improvement/Using-a-Multimeter>
32. <https://circuitdigest.com/tutorial/different-types-of-sensors-and-their-working>
33. <https://robu.in/wp-content/uploads/2019/09/Grove-Rotary-Angle-Sensor-User-Manual.pdf>
34. <https://www.elprocus.com/robot-sensor/>
35. <https://robu.in/ultrasonic-sensor-working-principle/>
36. <https://www.watelectronics.com/lcd-16x2/>
37. <https://www.automate.org/blogs/what-kinds-of-applications-are-best-for-stepper-motors>
38. [https://circuitdigest.com/article/servo-motor-working-and-basics#:~:text=Servo%20motor%20works%20on%20PWM,\(potentiometer\)%20and%20some%20gears.](https://circuitdigest.com/article/servo-motor-working-and-basics#:~:text=Servo%20motor%20works%20on%20PWM,(potentiometer)%20and%20some%20gears.)
39. <https://www.raspberrypi.com/documentation/computers/getting-started.html>
40. [GrovePi+Getting-started-Guide.pdf \(storage.googleapis.com\)](https://storage.googleapis.com/GrovePi+Getting-started-Guide.pdf)
41. [GrovePi Plus - Seeed Wiki \(seeedstudio.com\)](https://seeedstudio.com/GrovePi%20Plus%20-%20Seeed%20Wiki)
42. [Setting Up The Software for GrovePi \(seeedstudio.com\)](https://seeedstudio.com/Setting%20Up%20The%20Software%20for%20GrovePi)
43. [Grove - Ultrasonic Ranger - Seeed Wiki \(seeedstudio.com\)](https://seeedstudio.com/Grove%20-%20Ultrasonic%20Ranger%20-%20Seeed%20Wiki)
44. [Data Communications and Networking](https://seeedstudio.com/Data%20Communications%20and%20Networking)
45. <https://www.quora.com/How-does-the-Internet-of-Things-work-in-a-LAN-network.>
46. <https://www.f5.com/services/resources/glossary/multi-homing>
47. <https://behrtech.com/blog/wireless-iot-protocols-breaking-down-the-network-stack/>
- 48.
49. <https://www.iotcommunications.com/blog/types-of-iot-networks/>
50. <https://www.lairdconnect.com/resources/blog/all-about-iot>
51. <https://www.guru99.com/difference-ipv4-vs-ipv6.html>
52. <https://www.electronicshub.org/what-is-relay-and-how-it-works/>
53. <https://circuitglobe.com/relay.html>
54. [https://wiki.seeedstudio.com/Grove-Light Sensor/](https://wiki.seeedstudio.com/Grove-Light_Sensor/)
55. [https://wiki.seeedstudio.com/Grove-Sound Sensor/](https://wiki.seeedstudio.com/Grove-Sound_Sensor/)
56. [https://wiki.seeedstudio.com/Grove-LCD RGB Backlight/](https://wiki.seeedstudio.com/Grove-LCD_RGB_Backlight/)
57. <https://iotdesignpro.com/articles/different-types-of-wireless-communication-protocols-for-iot>
58. https://thingspeak.com/pages/learn_more
59. <https://knepublishing.com/index.php/Kne-Social/article/view/4128/8495>

60. <https://www.javatpoint.com/deep-learning-algorithms>
61. <https://www.guru99.com/deep-learning-tutorial.html>
62. https://www.tutorialspoint.com/python_deep_learning/index.htm
63. <https://www.datacamp.com/tutorial/tutorial-deep-learning-tutorial>
64. <https://www.javatpoint.com/gradient-descent-in-machine-learning>
65. <https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide>
66. <https://www.section.io/engineering-education/understanding-loss-functions-in-machine-learning/>
67. <https://analyticsindiamag.com/a-beginners-guide-to-cross-entropy-in-machine-leaing/#:~:text=The%20average%20number%20of%20bits,of%20actual%20and%20expected%20results.>
68. <https://www.v7labs.com/blog/overfitting#:~:text=It%20is%20a%20common%20pitfall,the%20noise%20and%20random%20fluctuations.>
69. <https://www.javatpoint.com/regularization-in-machine-learning>
70. <https://www.simplilearn.com/tutorials/deep-learning-tutorial/tensorflow-2>
71. <https://www.cs.ryerson.ca/~aharley/vis/>
72. <https://setosa.io/ev/image-kernels/>
73. <https://towardsdatascience.com/understand-transposed-convolutions-and-build-your-own-transposed-convolution-layer-from-scratch-4f5d97b2967>
74. <https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm>
75. <https://medium.com/voice-tech-podcast/text-classification-using-cnn-9ade8155dfb9>
76. "Release Notes for Intel Distribution of OpenVINO toolkit 2022". March 2022.
77. "OpenVINO Toolkit: Welcome to OpenVINO".
78. "Introduction to Intel Deep Learning Deployment Toolkit – OpenVINO Toolkit".
79. Wilbur, Marcia. "Use the Model Downloader and Model Optimizer for the Intel® Distribution of OpenVINO™ Toolkit on Raspberry Pi**".
80. Agrawal, Vasu (2019). Ground Up Design of a Multi-modal Object Detection System (PDF) (MSc). Carnegie Mellon University Pittsburgh, PA. Archived (PDF) from the original on 26 January 2020.
81. Driaba, Alexander; Gordeev, Aleksei; Klyachin, Vladimir (2019). "Recognition of Various Objects from a Certain Categorical Set in Real Time Using Deep Convolutional Neural Networks" (PDF). Institute of Mathematics and Informational Technologies Volgograd State University. Archived (PDF) from the original on 26 January 2020. Retrieved 26 January 2020
82. Nanjappa, Ashwin (31 May 2019). Caffe2 Quick Start Guide: Modular and scalable deep learning made easy. Packt. pp. 91–98. ISBN 978-1789137750.
83. <https://www.sap.com/india/about/company/what-is-sap.html>
84. <https://www.investopedia.com/terms/e/erp.asp>
85. <https://www.g2.com/categories/erp-systems>

86. https://learning.sap.com/learning-journey/get-started-with-abap-programming-on-sap-btp/understanding-the-basic-features-of-abap_c0e5346f-a136-4b9f-a167-903
87. <https://www.sap.com/documents/2018/08/7a62516c-157d-0010-87a3-c30de2ffd8ff.html>
88. [29.https://open.sap.com/courses/mm4h2](https://open.sap.com/courses/mm4h2)
89. [30.https://www.udemy.com/topic/sap-mm/](https://www.udemy.com/topic/sap-mm/)
90. [31.https://blogs.sap.com/2008/05/09/sap-project-system-a-ready-reference-part-1](https://blogs.sap.com/2008/05/09/sap-project-system-a-ready-reference-part-1)
91. [32.https://training.sap.com/content/sap-training-hana](https://training.sap.com/content/sap-training-hana)
92. [33.https://learning.sap-press.com/abap](https://learning.sap-press.com/abap)
93. [34.https://data-flair.training/blogs/sap-hana-vs-sap-s-4-hana/](https://data-flair.training/blogs/sap-hana-vs-sap-s-4-hana/)
94. [35.https://axxis-consulting.com/what-is-sap-c4hana/](https://axxis-consulting.com/what-is-sap-c4hana/)