

2021

# Artificial Intelligence

## Table of Contents

Table of Contents .....	2
Learning Outcome .....	6
<b>Chapter 1: What is AI and ML? .....</b>	<b>7</b>
1.1 Artificial Intelligence Market Trends .....	7
1.2 AI Market Investment.....	8
1.3 AI Market Opportunities .....	9
1.4 AI Career Opportunities .....	10
1.5 Introduction to Artificial Intelligence .....	11
1.6 Introduction to Machine Learning.....	18
1.7 Evolution of Artificial Intelligence .....	22
1.8 PRACTICAL : Microsoft AI Demos – Text Analytics .....	25
1.9 PRACTICAL : Language Understanding .....	28
1.10 PRACTICAL : Image Recognition on Web Browser .....	32
<b>Chapter 2 : Linux Kernel and Data Analysis with Python.....</b>	<b>41</b>
2.1 Introduction to Linux .....	41
2.2 Installation of Ubuntu .....	48
2.3 Linux Commands.....	60
2.4 Shell Scripting.....	67
2.5 Working on different text editors: nano, vi.....	70
2.6 Managing Linux Files .....	77
2.7 What is Data Analysis?.....	88
2.8 Anaconda Software and Introduction to Python.....	94
2.9 Installing Anaconda .....	98
2.10 Variables in Python.....	121
2.11 Advanced Datatypes in Python.....	122
2.12 Functions & Methods in Python .....	129
2.13 Condition & Loop in Python .....	133
2.14 Strings & Methods .....	141
2.15 Formatting in Python.....	145
2.16 Python Libraries.....	147
2.17 User-Defined Library.....	149
<b>Chapter 3: Data Analysis with Python.....</b>	<b>152</b>

3.1 Numpy Library .....	152
3.2 Creating scalars in numpy .....	158
3.3 Creating Vector in numpy .....	159
3.4 Creating Matrix in numpy .....	160
3.5 Matrix Multiplication in Numpy .....	160
3.6 NumPy Statistical Functions .....	161
3.7 Percentile & Interquartile in Numpy .....	165
3.8 Array Broadcasting in Numpy .....	167
3.9 Data Creation.....	168
3.10 What is Data Visualization? .....	169
3.11 Graph Plotting in Python using matplotlib .....	173
3.12 Figures and subplots .....	174
3.13 Colors, Markers and line styles.....	177
3.14 Matplotlib Configuration .....	180
3.15 Line Plot, Bar Plot, Scatter Plot.....	181
3.16 Ticks, Labels and Legends, subplots .....	184
3.17 Histograms and Binning.....	189
3.18 Text and Annotations .....	191
3.19 Three-Dimensional Plotting in Matplotlib .....	193
3.20 Visualizing with NumPy .....	196
3.21 Data Manipulation with Pandas .....	198
3.22 Introducing Pandas Objects.....	203
3.23 Creating Series from simple datatypes .....	206
3.24 Data Storage Formats in Pandas.....	209
3.25 CSV file and JSON file.....	210
3.26 Reading data from files.....	214
3.27 Groupby Methods .....	224
3.28 Pivot Tables .....	227
3.29 Pandas Plotting.....	231
<b>Chapter 4 : Building Machine Learning Models.....</b>	<b>238</b>
4.1 Machine Learning Basics.....	238
4.2 Techniques of ML .....	242
4.3 Scikit Learn library overview .....	245

4.4 Regression vs Classification .....	247
4.5 Least Square Method .....	249
4.6 Azure ML No Code Platform .....	258
4.7 Setting up an account in Azure ML Studio and Creating Workspace.....	260
4.8 Training a ML model in Azure Studio.....	263
4.9 Create a Regression model with Azure ML Studio .....	276
4.10 Logistic Regression .....	289
4.11 Logistic Regression model with ML Studio .....	293
4.12 GridSearchCV for Hyperparameter Tuning.....	302
4.13 Decision Trees.....	308
4.14 Gini Impurity or Entropy .....	310
4.15 Linear SVM Classification.....	312
4.16 Different distance methods .....	320
4.17 Geometric Intuition of K-NN.....	323
4.18 Probability Theory .....	327
4.19 Naïve Bayes Classifier Algorithm.....	333
4.20 Bag of Words Approach.....	336
4.21 Lexicons for Sentiment Analysis .....	342
4.22 Unsupervised Learning .....	343
4.23 Clustering .....	345
4.24 Agglomerative Clustering.....	347
4.25 DBSCAN.....	348
4.26 K-means clustering .....	349
<b>Chapter 5: Building Deep Learning Models .....</b>	<b>352</b>
5.1 Deep Learning Basics.....	352
5.2 Concept of Neural Networks .....	355
5.3 Overfitting in Deep Learning .....	359
5.4 Computer Vision Basics.....	366
5.5 Image Processing .....	370
5.6 Facial Detection .....	374
5.7 Convolutional Neural Networks.....	379
5.8 What is transfer learning? .....	385
5.9 Optical Character Recognition .....	391

Reference .....	396
-----------------	-----

**This course booklet has been designed by Edunet Foundation for the Tech-Saksham programme in partnership with Microsoft and SAP**

## Learning Outcome

After completing this handbook, learner will be able to

- Demonstrate fundamental understanding of the history of artificial intelligence and its foundations.
- Apply the basic principles, models, and algorithms of AI to recognize, model, and solve problems in the analysis and design of information systems.
- Analyze the structures and algorithms of a selection of techniques related to machine learning and Artificial Intelligence.
- Able to design and implement various machine learning algorithms in a range of real-world applications.
- Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and unsupervised learning.
- Be able to identify new application requirements in the field of computer vision using Deep Learning.

# Chapter 1: What is AI and ML?

## Learning Outcomes:

- Understand the AI market trends, investments and career opportunities
- Understand fundamental concepts of Artificial Intelligence and Machine Learning
- Create a Machine Learning model to differentiate between Images

## 1.1 Artificial Intelligence Market Trends

AI has entered a new stage to become fully commercialized, exerting different impacts on players of traditional industries and driving changes in the ecosystems of these industries. Such changes are mainly seen at three levels.

### *Enterprise change*

AI is engaged in the management and production processes of the enterprise, with a trend of being increasingly commercialized, and some enterprises have realized relatively mature intelligent applications. These enterprises have been able to collect and make use of user information from multiple dimensions via various technological means and provide consumers with pertinent products and services, at the same time satisfy their potential needs through insights into development trends gained via data optimization.



#### 1. Enterprise change

Sales      Security      Anti-fraud      HR management      Marketing      Personal assistant      Smart tools

### *Industry change*

The change brought by AI would drive fundamental changes in the relationship of upstream and downstream sectors on the traditional industry chain. The engagement of AI has expanded the types of upstream products providers, and users may also shift from individual consumers to enterprise consumers.



## 2. Industry change

Finance	Healthcare	Education	Autonomous driving	Retail	Manufacturing
Digital government	Media	Legal	Agriculture	Logistics	Oil & gas

### Labour change

The application of new technologies such as AI is enhancing the efficiency of information use and reducing the number of employees. In addition, the wider use of robots would also replace labours in repetitive tasks and increase the percentage of technological and management personnel, bringing changes in the labour structures of enterprises.



## 3. Labor change

Augmented reality	Gesture recognition	Robotics	Emotion recognition
-------------------	---------------------	----------	---------------------

## 1.2 AI Market Investment

As the capital market deepens its understanding of AI, market investment in AI is maturing and returning to reason. During the past five years, China's investment in AI grew rapidly, with a total investment of RMB45 billion in 2015 - the starting year of China's AI development, and investment frequency continued to increase in 2016 and 2017. The first half of 2019 has seen a total investment of over RMB47.8 billion in China's AI sector with great achievements.

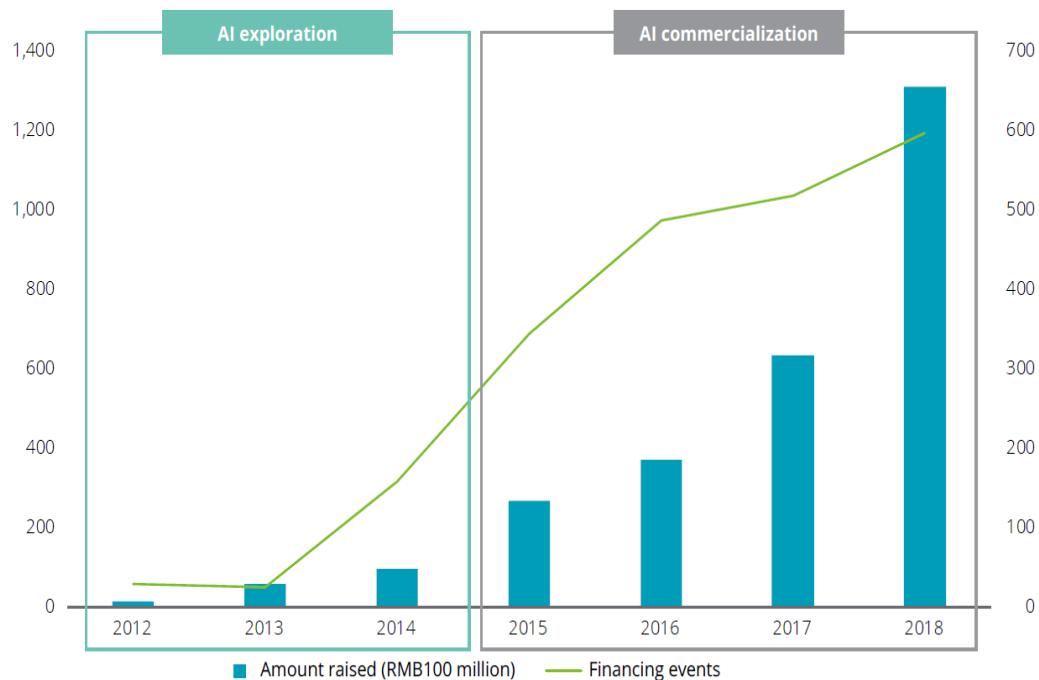


Image: Changes of AI investment and financing

Reference: [Public information, Deloitte Research](#)

## 1.3 AI Market Opportunities

Investment and financing data in recent years show that investment frequencies and amounts raised in business service, robotics, healthcare, industry solutions, basic components and finance are all higher than those in other sectors. From enterprise perspective, those with a top global team, financial strength and high-tech gene are more favoured by secondary market investors. From industry perspective, however, new retail, autonomous driving, healthcare and education, all easy to deploy, indicate more opportunities, and companies engaged in such sectors could see more investment opportunities.

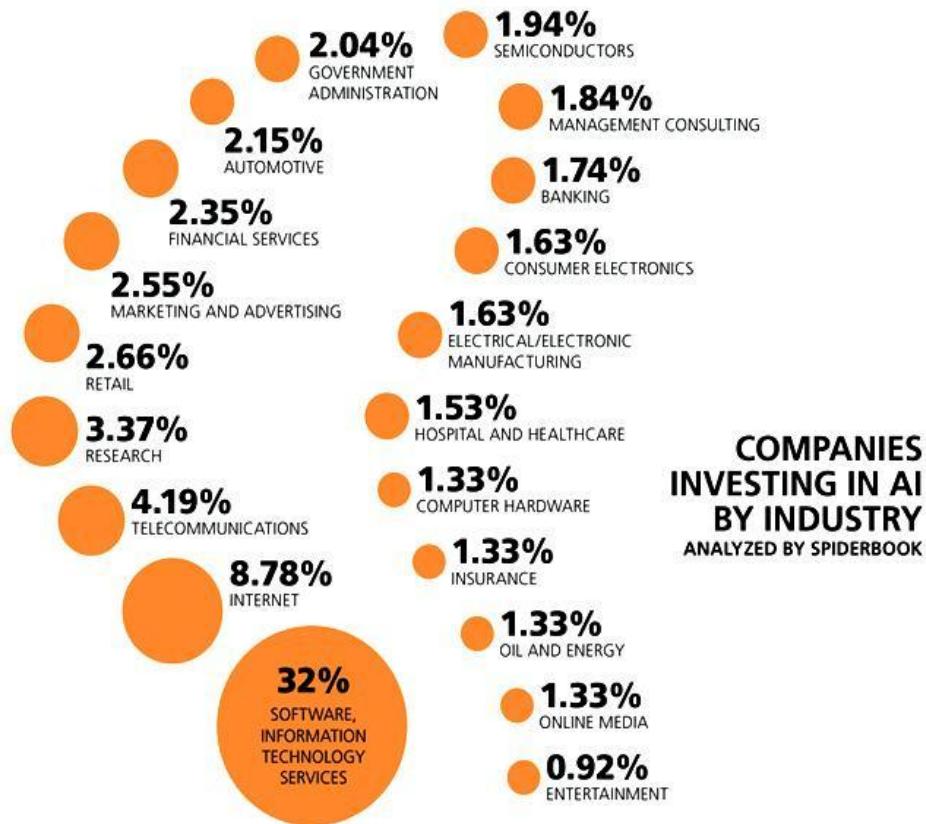


Image: Market opportunities and career in AI

Reference: <http://www.oreilly.com/data/free/the-new-artificial-intelligence-market.csp>

## 1.4 AI Career Opportunities

The World Economic Forum's "The Future of Jobs 2018" aims to base this debate on facts rather than speculation. By tracking the acceleration of technological change as it gives rise to new job roles, occupations and industries, the report evaluates the changing contours of work in the Fourth Industrial Revolution.

One of the primary drivers of change identified is the role of emerging technologies, such as artificial intelligence (AI) and automation. The report seeks to shed more light on the role of new technologies in the labour market, and to bring more clarity to the debate about how AI could both create and limit economic opportunity. With 575 million members globally, LinkedIn's platform provides a unique vantage point into global labour-market developments, enabling us to support the Forum's examination of the trends that will shape the future of work.

Our analysis uncovered two concurrent trends: the continued rise of tech jobs and skills, and, in parallel, a growth in what we call "human-centric" jobs and skills. That is, those that depend on intrinsically human qualities.

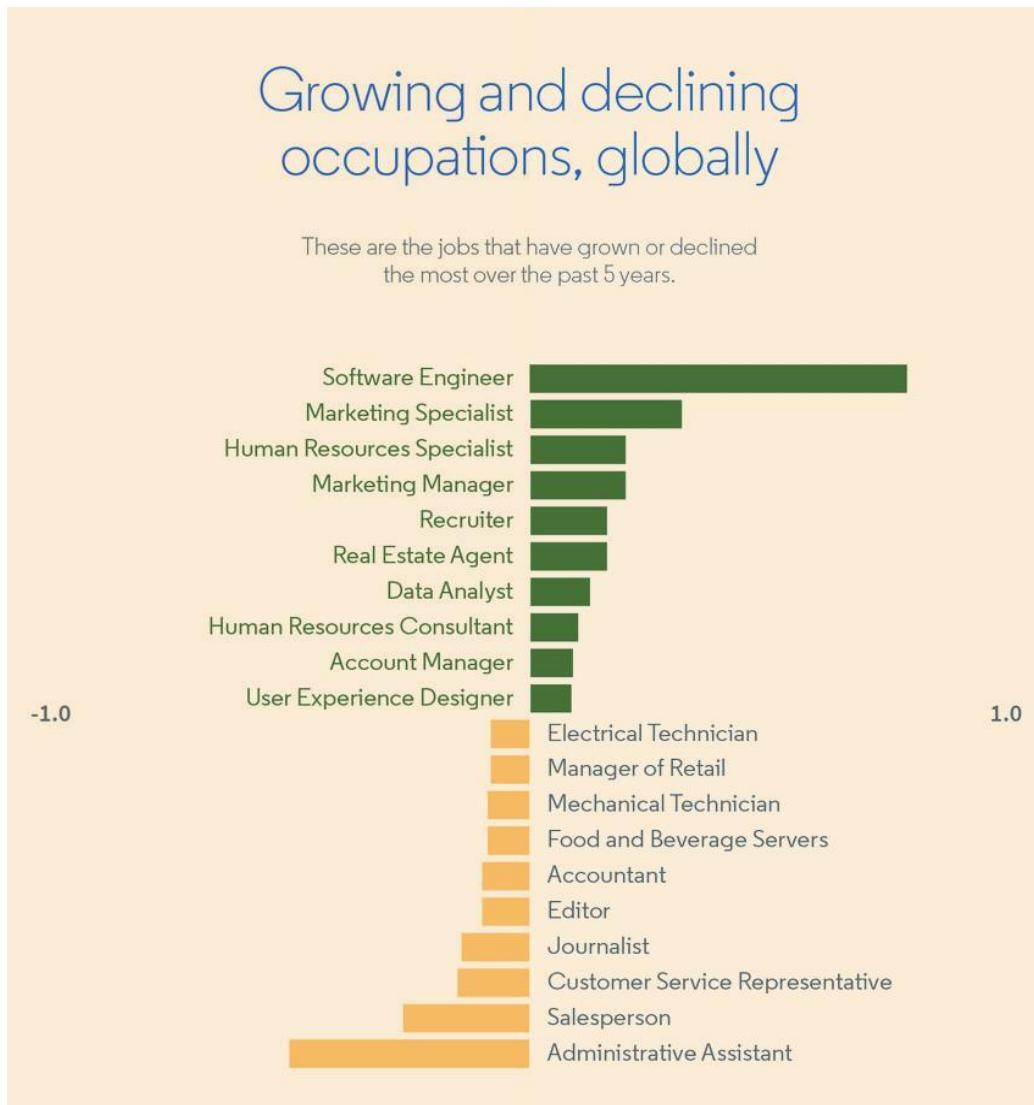


Image: career opportunities in AI

Reference: <https://www.weforum.org/agenda/2018/09/artificial-intelligence-shaking-up-job-market/>

Tech jobs like software engineers and data analysts, along with technical skills such as cloud computing, mobile application development, software testing and AI, are on the rise in most industries and across all regions. But several highly “automatable” jobs fall into the top 10 most declining occupations – i.e., jobs that have seen the largest decreases in share of hiring over the past five years. These occupations include administrative assistants, customer service representatives, accountants, and electrical/mechanical technicians, many of which depend on more repetitive tasks.

## 1.5 Introduction to Artificial Intelligence

### Definitions

**Artificial intelligence (AI)** is the intelligence exhibited by machines or software. It is also the name of the academic field of study which studies how to create computers and computer software that are capable of intelligent behaviour.

**Artificial intelligence (AI)** is an area of computer science that emphasizes the creation of intelligent machines that work and react like humans. Some of the activities computers with artificial intelligence are designed for include: Speech recognition, Learning, Planning, Problem solving. In this topic we shall discuss the following subjects; Deep learning, Machine learning, Computer Programming, Medical field.

### *Introduction of AI*

Artificial intelligence is a branch of computer science that aims to create intelligent machines. It has become an essential part of the technology industry. Research associated with artificial intelligence is highly technical and specialized. The core problems of artificial intelligence include programming computers for certain traits such as: Knowledge, Reasoning, Problem solving, Perception, Learning, Planning, and Ability to manipulate and move objects. Knowledge engineering is a core part of AI research.

Machines can often act and react like humans only if they have abundant information relating to the world. Artificial intelligence must have access to objects, categories, properties and relations between all of them to implement knowledge engineering. Initiating common sense, reasoning and problem-solving power in machines is a difficult and tedious approach. Machine learning is another core part of AI.

Learning without any kind of supervision requires an ability to identify patterns in streams of inputs, whereas learning with adequate supervision involves classification and numerical regressions. Classification determines the category an object belongs to and regression deals with obtaining a set of numerical input or output examples, thereby discovering functions enabling the generation of suitable outputs from respective inputs.

Mathematical analysis of machine learning algorithms and their performance is a well-defined branch of theoretical computer science often referred to as computational learning theory. Machine perception deals with the capability to use sensory inputs to deduce the different aspects of the world, while computer vision is the power to analyse visual inputs with a few sub- problems such as facial, object and gesture recognition. Robotics is also a major field related to AI. Robots require intelligence to handle tasks such as object manipulation and navigation, along with sub-problems of localization, motion planning and mapping.

### Classification of AI

**Artificial Intelligence**, also called **(AI)**, is an imitation of the human intelligence system, and the processes involved are carried out by a computing device (a computer, robot, smartphone or any such device). It includes the processes of **acquiring information, validating assigned rules, and debugging itself** by the intelligent software and **executing actions** by the hardware installed with it.

Let's simplify the definition of AI; a machine, robot or android that has an intelligent system like a human brain which can **sense, reason, act & adapt** according to the operational instructions. It works based on the data, stored as well as collected, and configures itself with real-time instructions.

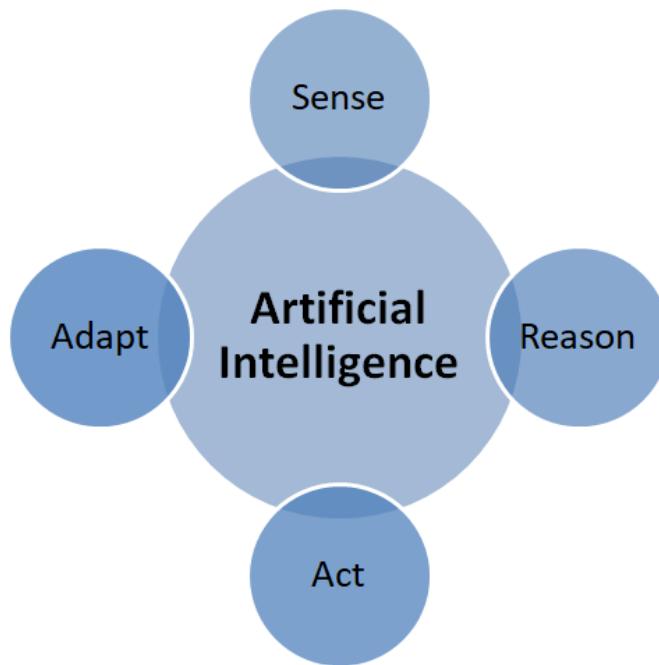
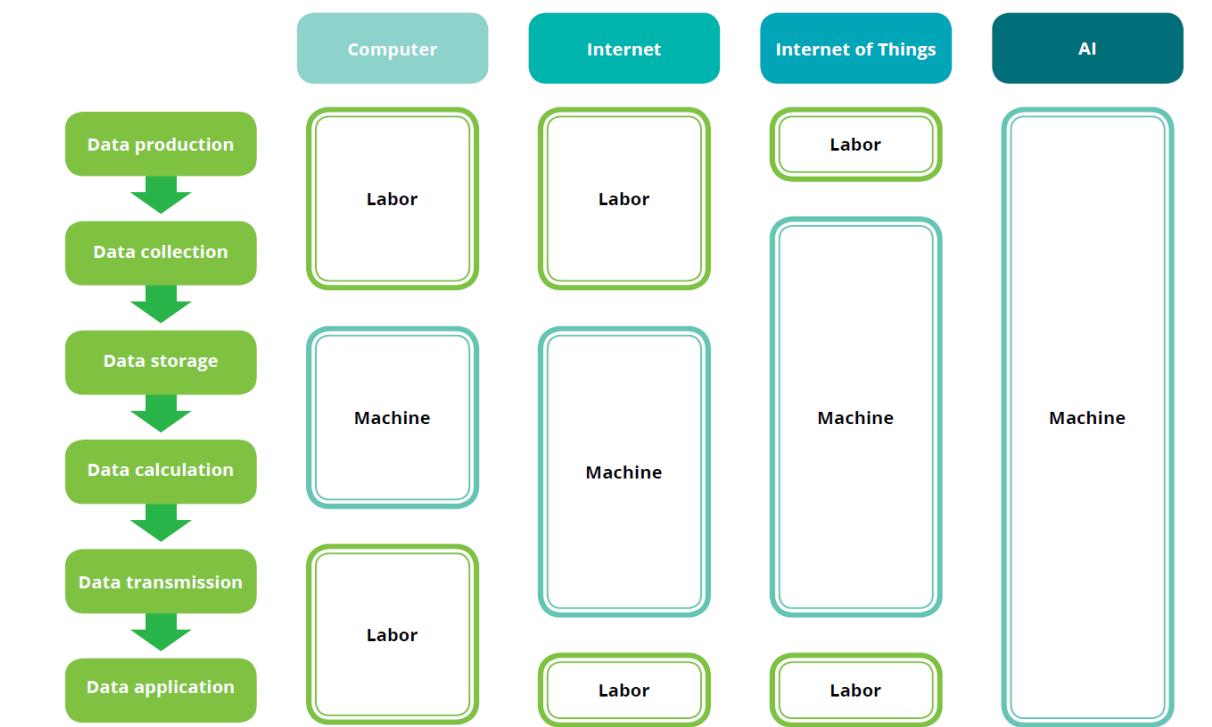


Image: Basic Block diagram of AI

- Sense: It's one of the properties of AI. It not only identifies worthy materials & objects but also recognizes real-time operational activities. The sensors or sensing devices can trace out & quickly differentiate between wrong & correct objects.
- Reason: This property works like a human brain does to complete a task successfully. It understands, judges and prepares to execute. Reason enables AI to deal with internal & external properties of resources like condition, time-frame, behavior, other parameters of entities involved during conversion of the task.
- Act: This is a decisive property that enables to execute an action or send instructions to others to execute the action instead. The act is the part where the functionality & operational activities are directly proportionate.

- Adapt: It's the property that works with the highest intelligence. Like the way, a human brain remembers the result of any past event. Its re-trains, debugs and even find out uncovered properties to put the operation more accurate. It remembers the last events & manages the functionalities accordingly.

For AI data is an essential element that underpins AI's underlying logic. Without data, data processing for AI will not be possible. With data mining's cleaning, integration, reduction and other pre-treatment means, AI could have adequate data for learning. As AI technologies iterate, the production, collection, storage, calculation, transmission and application of data will all be completed by machines.



## Applications of AI

Here we have some of the Artificial Intelligence Applications in real world.



Image: Applications of Artificial Intelligence

Reference: <https://techvidvan.com/tutorials/artificial-intelligence-applications/>

### **1. Healthcare**

One of the foremost deep-lying impacts which AI has created is within the Healthcare space. A device, as common as a Fitbit or an iWatch, collects a lot of data like the sleep patterns of the individual, the calories burnt by him, heart rate and a lot more which can help with early detection, personalization, even disease diagnosis.

This device, when powered with AI can easily monitor and notify abnormal trends. This can even schedule a visit to the closest Doctor by itself and therefore, it's also of great help to the doctors who can get help in making decisions and research with AI. It has been used to predict ICU transfers, improve clinical workflows and even pinpoint a patient's risk of hospital-acquired infections.

### **2. Automobile**

At this stage where automobiles changing from an engine with a chassis around it to a software-controlled intelligent machine, the role of AI cannot be underestimated.

The goal of self-driving cars, during which Autopilot by Tesla has been the frontrunner, takes up data from all the Tesla's running on the road and uses it in machine learning algorithms. The assessment of both chips is later matched by the system and followed if the input from both is the same.

### **3. Banking and Finance**

One of the early adopters of Artificial Intelligence is the Banking and Finance Industry.

Features like AI bots, digital payment advisers and biometric fraud detection mechanisms cause higher quality of services to a wider customer base.

The adoption of AI in banking is constant to rework companies within the industry, provide greater levels useful and more personalized experiences to their customers,

reduce risks as well as increase opportunities involving financial engines of our modern economy.

#### ***4. Surveillance***

AI has made it possible to develop face recognition Tools which may be used for surveillance and security purposes. As a result, this empowers the systems to monitor the footage in real-time and can be a pathbreaking development in regards to public safety.

Manual monitoring of a CCTV camera requires constant human intervention so they're prone to errors and fatigue. AI-based surveillance is automated and works 24/7, providing real-time insights. According to a report by the Carnegie Endowment for International Peace, a minimum of 75 out of the 176 countries are using AI tools for surveillance purposes.

Across the country, 400 million CCTV cameras are already in situ, powered by AI technologies, primarily face recognition.

#### ***5. Social Media***

Social Media is not just a platform for networking and expressing oneself. It subconsciously shapes our choices, ideologies, and temperament.

This is due to the synthetic Intelligence tools which work silently within the background, showing us posts that we "might" like and advertising products that "might" be useful based on our search and browsing history.

This helps with social media advertising because of its unprecedented ability to run paid ads to platform users based on highly granular demographic and behavioural targeting.

#### ***6. Entertainment***

The show business, with the arrival of online streaming services like Netflix and Amazon Prime, relies heavily on the info collected by the users.

This helps with recommendations based upon the previously viewed content. This is done not only to deliver accurate suggestions but also to create content that would be liked by a majority of the viewers.

With new contents being created every minute, it is very difficult to classify them and making them easier to search. AI tools analyse the contents of videos frame by frame and identify objects to feature appropriate tags. AI is additionally helping media companies to form strategic decisions.

#### ***7. Education***

In the education sector also, there are a number of problems which will be solved by the implementation of AI.

A few of them being automated marking software, content retention techniques and suggesting improvements that are required. This can help the teachers monitor not just the academic but also the psychological, mental and physical well-being of the students but also their all-round development.

This would also help in extending the reach of education to areas where quality educators can't be present physically.

### ***8. Space Exploration***

AI systems are being developed to scale back the danger of human life that venture into the vast realms of the undiscovered and unravelled universe which is a very risky task that the astronauts need to take up.

As a result, unmanned space exploration missions just like the Mars Rover are possible due to the utilization of AI. It has helped us discover numerous exoplanets, stars, galaxies, and more recently, two new planets in our very own system.

NASA is also working with AI applications for space exploration to automate image analysis and to develop autonomous spacecraft that would avoid space debris without human intervention, create communication networks more efficient and distortion-free by using an AI-based device.

### ***9. Gaming***

In the gaming industry also, computer game Systems powered by AI is ushering us into a replacement era of immersive experience in gaming. It serves to enhance the game-player experience instead of machine learning or deciding. AI has also been playing a huge role in creating video games and making it more tailored to players' preferences.

### ***10. Robotics***

With increasing developments within the field of AI, robots are becoming more efficient in performing tasks that earlier were too complex.

AI in robotics helps the robots to learn the processes and perform the tasks with complete autonomy, without any human intervention. This is because robots are designed to perform repetitive tasks with utmost precision and increased speed.

AI has been introducing flexibility and learning capabilities in previously rigid applications of robots. These benefits are expected to reinforce the market growth.

### ***11. Agriculture***

Artificial Intelligence is changing the way we do one among our most primitive and basic professions which are farming. The use of AI in agriculture are often attributed to agriculture robots, predictive analysis, and crop and soil monitoring.

In addition, drones are also used for spraying insecticides and detecting weed formation in large farms. This is getting to help firms like Blue River Technologies, better manage the farms.

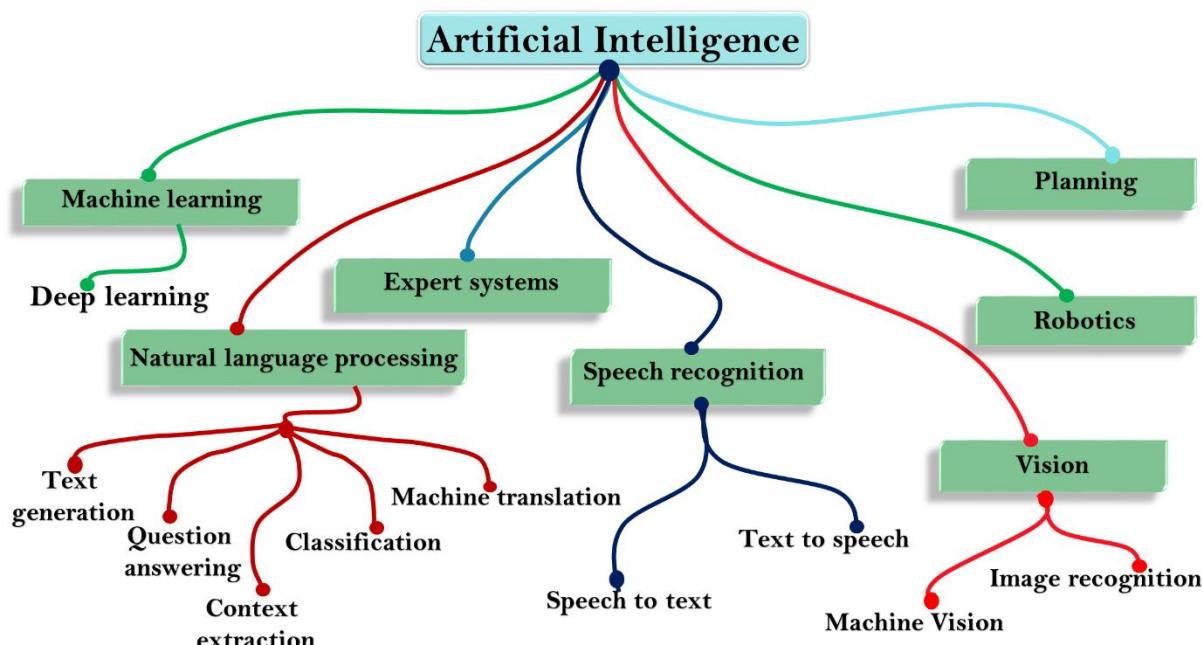
AI has also enhanced crop production and improved real-time monitoring, harvesting, processing and marketing.

## 12. E-Commerce

This is one of the Artificial Intelligence Applications that's found to be widely used. Different departments of E-commerce including logistics, predicting demand, intelligent marketing, better personalization, use of chatbots, etc. are being disrupted by AI. The E-Commerce industry, a prominent player being Amazon is one among the primary industries to embrace AI. This may experience a good use of AI with time.

E-commerce retailers are increasingly turning towards chatbots or digital assistants to supply 24x7 support to their online buyers.

## AI Subdomains



Reference - <https://static.javatpoint.com/tutorial/ai/images/subsets-of-ai.png>

## 1.6 Introduction to Machine Learning

Machine learning is the study of computer algorithms that improve automatically through experience and has been central to AI research since the field's inception. Unsupervised learning is the ability to find patterns in a stream of input. Supervised learning includes both classification and numerical regression.

Classification is used to determine what category something belongs in, after seeing a number of examples of things from several categories. Regression is the attempt to produce a function that describes the relationship between inputs and outputs and predicts how the outputs should change as the inputs change. In reinforcement learning the agent is rewarded for good responses and punished for bad ones.

## Introduction of ML

Machine learning pipeline helps to automate ML Workflow and enable the sequence data to be transformed and correlated together in a model to analyzed and achieve outputs. ML pipeline is constructed to allow the flow of data from raw data format to some valuable information. It provides a mechanism to build a Multi-ML parallel pipeline system to examine different ML methods' outcomes.

The Objective of the Machine learning pipeline is to exercise control over the ML model. A well-planned pipeline helps to makes the implementation more flexible. It is like having an overview of a code to pick the fault and replace them with the correct code.

A pipeline consists of several stages. Each stage of a pipeline is fed with the data processed from its preceding stage, i.e., the output of a processing unit supplied as an input to the next step. Machine Learning Pipeline consists of four main stages as Pre-processing, Learning, Evaluation, and Prediction.

### *1. Pre-processing*

Data preprocessing is a Data Mining technique that involves transferring raw data into an understandable format. Real-world data is usually incomplete, inconsistent, and lacks certain behaviors or trends, most likely to contain many inaccuracies. The process of getting usable data for a Machine Learning algorithm follows steps such as Feature Extraction and Scaling, Feature Selection, Dimensionality reduction, and sampling. The product of Data Pre-processing is the final dataset used for training the model and testing purposes.

### *2. Learning*

A learning algorithm is used to process understandable data to extract patterns appropriate for application in a new situation. In particular, the aim is to utilize a system for a specific input-output transformation task. For this, choose the best-performing model from a set of models produced by different hyperparameter settings, metrics, and cross-validation techniques.

### *3. Evaluation*

To Evaluate the Machine Learning model's performance, fit a model to the training data, and predict the labels of the test set. Further, count the number of wrong predictions on the test dataset to compute the model's prediction accuracy.

#### 4. Prediction

The model's performance to determine the outcomes of the test data set was not used for any training or cross-validation activities.

## Machine Learning Applications

Machine learning is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that which makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect. We probably use a learning algorithm dozen of time without even knowing it. Applications of Machine Learning include:

### a. Web Search Engine

One of the reasons why search engines like google, Bing etc. work so well is because the system has learnt how to rank pages through a complex learning algorithm.

### b. Photo tagging Applications

Be it Facebook or any other photo tagging application, the ability to tag friends makes it even more happening. It is all possible because of a face recognition algorithm that runs behind the application.

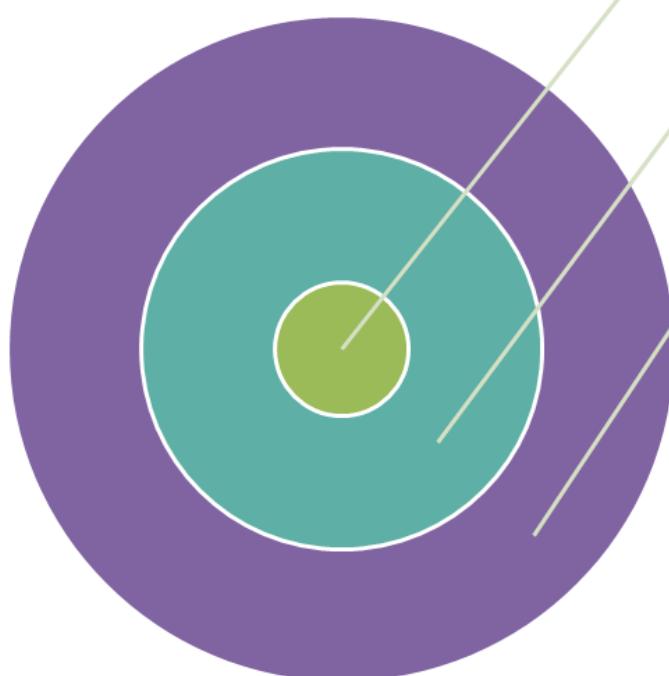
### c. Spam Detector

Our mail agent like Gmail or Hotmail does a lot of hard work for us in classifying the mails and moving the spam mails to spam folder. This is again achieved by a spam classifier running in the back end of mail application.

## The key difference between AI and ML are:

ARTIFICIAL INTELLIGENCE	MACHINE LEARNING
AI stands for Artificial intelligence, where intelligence is defined acquisition of knowledge intelligence is defined as an ability to acquire and apply knowledge.	ML stands for Machine Learning which is defined as the acquisition of knowledge or skill
The aim is to increase chance of success and not accuracy.	The aim is to increase accuracy, but it does not care about success
It works as a computer program that does smart work	It is a simple concept machine takes data and learn from data.

The goal is to simulate natural intelligence to solve complex problem	The goal is to learn from data on certain task to maximize the performance of machine on this task.
AI is decision making.	ML allows system to learn new things from data.
It leads to develop a system to mimic human to respond behave in a circumstance.	It involves in creating self-learning algorithms.
AI will go for finding the optimal solution.	ML will go for only solution for that whether it is optimal or not.
AI leads to intelligence or wisdom.	ML leads to knowledge.



**Deep learning**-A subset of ML based on neural networks that permit a machine to train itself to perform a task

**Machine Learning**- A subset of AI that includes techniques that enables machines to improve at task with experience

**Artificial Intelligence**-Any technique that enables computers to mimic human intelligence

Image: Artificial Intelligence

### ***Artificial intelligence***

*Artificial intelligence* (AI) is the capability of a computer to imitate intelligent human behaviour. Through AI, machines can analyse images, comprehend speech, interact in natural ways, and make predictions using data.

### ***AI concepts Algorithm***

An *algorithm* is a sequence of calculations and rules used to solve a problem or analyse a set of data. It is like a flow chart, with step-by-step instructions for questions to ask, but written in math and programming code. An algorithm may describe how to determine whether a pet is a cat, dog, fish, bird, or lizard. Another far more complicated algorithm may describe how to identify a written or spoken language, analyse its words, translate them into a different language, and then check the translation for accuracy.

### ***Machine learning***

*Machine learning* (ML) is an AI technique that uses mathematical algorithms to create predictive models. An algorithm is used to parse data fields and to "learn" from that data by using patterns found within it to generate models. Those models are then used to make informed predictions or decisions about new data. The predictive models are validated against known data, measured by performance metrics selected for specific business scenarios, and then adjusted as needed. This process of learning and validation is called *training*. Through periodic retraining, ML models are improved over time.

### ***Deep learning***

*Deep learning* is a type of ML that can determine for itself whether its predictions are accurate. It also uses algorithms to analyze data, but it does so on a larger scale than ML. Deep learning uses artificial neural networks, which consist of multiple layers of algorithms. Each layer looks at the incoming data, performs its own specialized analysis, and produces an output that other layers can understand. This output is then passed to the next layer, where a different algorithm does its own analysis, and so on.

With many layers in each neural network-and sometimes using multiple neural networks-a machine can learn through its own data processing. This requires much more data and much more computing power than ML.

## **1.7 Evolution of Artificial Intelligence**

Artificial Intelligence is not a new word and not a new technology for researchers. This technology is much older than you would imagine. Even there are the myths of Mechanical men in Ancient Greek and Egyptian Myths. Following are some milestones in the history of AI which defines the journey from the AI generation to till date development. [8]

## **History of Artificial Intelligence**

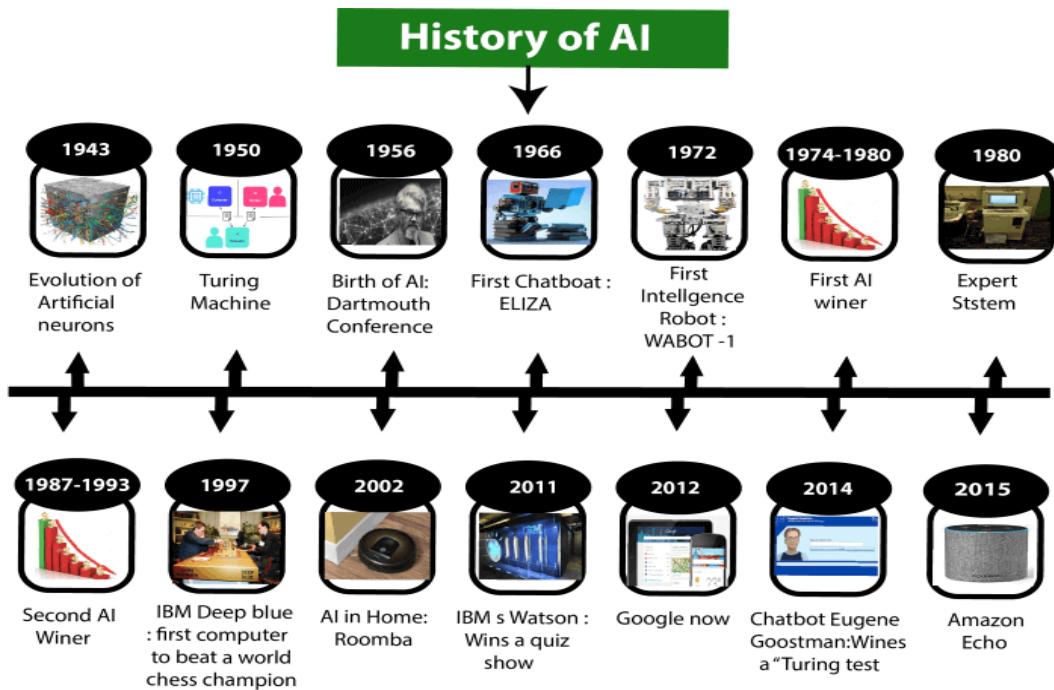


Image: History of AI

Reference: <https://www.javatpoint.com/history-of-artificial-intelligence>

### **Maturation of Artificial Intelligence (1943-1952)**

**Year 1943:** The first work which is now recognized as AI was done by Warren McCulloch and Walter Pitts in 1943. They proposed a model of artificial neurons.

**Year 1949:** Donald Hebb demonstrated an updating rule for modifying the connection strength between neurons. His rule is now called Hebbian learning.

**Year 1950:** The Alan Turing who was an English mathematician and pioneered Machine learning in 1950. Alan Turing publishes "Computing Machinery and Intelligence" in which he proposed a test. The test can check the machine's ability to exhibit intelligent behavior equivalent to human intelligence, called a Turing test.

### **The birth of Artificial Intelligence (1952-1956)**

**Year 1955:** An Allen Newell and Herbert A. Simon created the "first artificial intelligence program" which was named as "Logic Theorist". This program had proved 38 of 52 Mathematics theorems, and find new and more elegant proofs for some theorems.

**Year 1956:** The word "Artificial Intelligence" first adopted by American Computer scientist John McCarthy at the Dartmouth Conference. For the first time, AI coined as an academic field. At that time high-level computer languages such as FORTRAN, LISP, or COBOL were invented. And the enthusiasm for AI was very high at that time.

### ***The golden years-Early enthusiasm (1956-1974)***

**Year 1966:** The researchers emphasized developing algorithms which can solve mathematical problems. Joseph Weinbaum created the first chatbot in 1966, which was named as ELIZA.

**Year 1972:** The first intelligent humanoid robot was built in Japan which was named as WABOT-1.

### ***The first AI winter (1974-1980)***

The duration between years 1974 to 1980 was the first AI winter duration. AI winter refers to the time period where computer scientist dealt with a severe shortage of funding from government for AI researches. During AI winters, an interest of publicity on artificial intelligence was decreased.

### ***A boom of AI (1980-1987)***

**Year 1980:** After AI winter duration, AI came back with "Expert System". Expert systems were programmed that emulate the decision-making ability of a human expert.

In the Year 1980, the first national conference of the American Association of Artificial Intelligence was held at Stanford University.

### ***The second AI winter (1987-1993)***

The duration between the years 1987 to 1993 was the second AI Winter duration. Again, Investors and government stopped in funding for AI research as due to high cost but not efficient result. The expert system such as XCON was very cost effective.

### ***The emergence of intelligent agents (1993-2011)***

**Year 1997:** In the year 1997, IBM Deep Blue beats world chess champion, Gary Kasparov, and became the first computer to beat a world chess champion.

**Year 2002:** for the first time, AI entered the home in the form of Roomba, a vacuum cleaner.

**Year 2006:** AI came in the Business world till the year 2006. Companies like Facebook, Twitter, and Netflix also started using AI.

### ***Deep learning, big data and artificial general intelligence (2011-present)***

**Year 2011:** In the year 2011, IBM's Watson won jeopardy, a quiz show, where it had to solve the complex questions as well as riddles. Watson had proved that it could understand natural language and can solve tricky questions quickly.

**Year 2012:** Google has launched an Android app feature "Google now", which was able to provide information to the user as a prediction.

**Year 2014:** In the year 2014, Chatbot "Eugene Goostman" won a competition in the infamous "Turing test."

**Year 2018:** The "Project Debater" from IBM debated on complex topics with two master debaters and also performed extremely well.

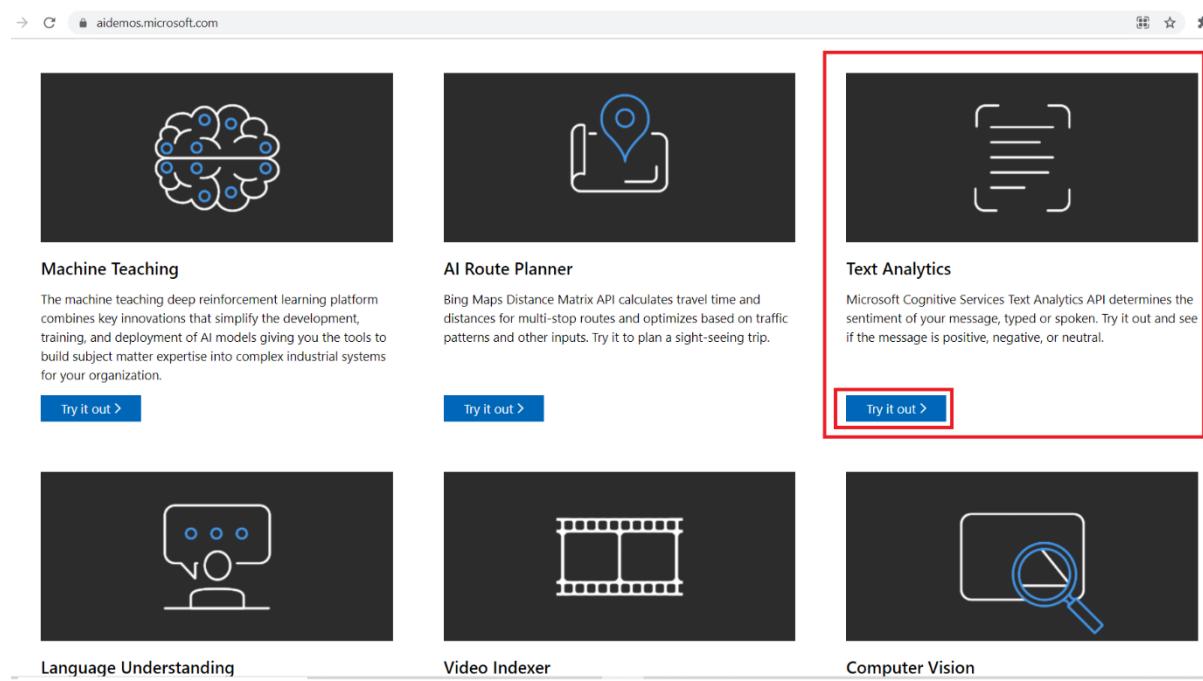
Google has demonstrated an AI program "Duplex" which was a virtual assistant and which had taken hairdresser appointment on call, and lady on other side didn't notice that she was talking with the machine.

Now AI has developed to a remarkable level. The concept of Deep learning, big data, and data science are now trending like a boom. Nowadays companies like Google, Facebook, IBM, and Amazon are working with AI and creating amazing devices. The future of Artificial Intelligence is inspiring and will come with high intelligence.

Let's get started with some practical demonstrations of some AI applications. In the very first practical which is based on Text Analytics which determines the sentiments of your message typed or spoken and predicts whether the message is Positive, Negative or Neutral.

## 1.8 PRACTICAL : Microsoft AI Demos – Text Analytics

1. Go to website <https://aidemos.microsoft.com/>
2. Select Text Analytics and click on “Try it out>” as shown in figure



The screenshot shows the Microsoft AI Demos website with six service cards:

- Machine Teaching**: A brain icon. Description: The machine teaching deep reinforcement learning platform combines key innovations that simplify the development, training, and deployment of AI models giving you the tools to build subject matter expertise into complex industrial systems for your organization. [Try it out >](#)
- AI Route Planner**: A map pin icon. Description: Bing Maps Distance Matrix API calculates travel time and distances for multi-stop routes and optimizes based on traffic patterns and other inputs. Try it out to plan a sightseeing trip. [Try it out >](#)
- Text Analytics**: A document icon. Description: Microsoft Cognitive Services Text Analytics API determines the sentiment of your message, typed or spoken. Try it out and see if the message is positive, negative, or neutral. [Try it out >](#) (This card is highlighted with a red border)
- Language Understanding**: A speech bubble icon.
- Video Indexer**: A film strip icon.
- Computer Vision**: A magnifying glass icon.

3. Enter any Message that you want for Text Analytics

aidemos.microsoft.com/text-analytics

## Text Analytics

- Step #1: Enter Your Message

Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius about nine times that of Earth.

[Next Step >](#)

- Step #2: Sentiment & Key Phrases

This API analyzes your text to identify the keywords and discern the sentiment.

[Next Step](#)

- Step #3: Entity Linking

This API ascertains which of the key words are entities and links them in Wikipedia.

[Next Step](#)

- Step #4: Bing Entity Search

This API provides a summary of relevant information in the form of a card for each entity

[Start Over](#) [Learn to code](#)



#### 4. Click on Next Step

aidemos.microsoft.com/text-analytics

## Text Analytics

- Step #1: Enter Your Message

Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius about nine times that of Earth.

[Next Step >](#)

- Step #2: Sentiment & Key Phrases

This API analyzes your text to identify the keywords and discern the sentiment.

[Next Step](#)

- Step #3: Entity Linking

This API ascertains which of the key words are entities and links them in Wikipedia.

[Next Step](#)

- Step #4: Bing Entity Search

This API provides a summary of relevant information in the form of a card for each entity

[Start Over](#) [Learn to code](#)



5. The API analyzes your Message to identify the Keywords and understands the Sentiment, you can see on the Screen the selected Key Words.  
Then Click on **Next Step**

C aidemos.microsoft.com/text-analytics

## Text Analytics

- Step #1: Enter Your Message

Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius about nine times that of Earth.

[Next Step](#)
- Step #2: Sentiment & Key Phrases

This API analyzes your text to identify the keywords and discern the sentiment.

[Next Step >](#)
- Step #3: Entity Linking

This API ascertains which of the key words are entities and links them in Wikipedia.

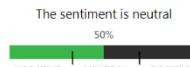
[Next Step](#)
- Step #4: Bing Entity Search

This API provides a summary of relevant information in the form of a card for each entity

[Start Over](#) [Learn to code >](#)

Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius about nine times that of Earth.

The sentiment is neutral  
50%



## 6. The API will select the key words as entities and link them to Wikipedia Then Click on **Next Step**

C aidemos.microsoft.com/text-analytics

## Text Analytics

- Step #1: Enter Your Message

Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius about nine times that of Earth.

[Next Step](#)
- Step #2: Sentiment & Key Phrases

This API analyzes your text to identify the keywords and discern the sentiment.

[Next Step](#)
- Step #3: Entity Linking

This API ascertains which of the key words are entities and links them in Wikipedia.

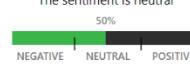
[Next Step >](#)
- Step #4: Bing Entity Search

This API provides a summary of relevant information in the form of a card for each entity

[Start Over](#) [Learn to code >](#)

Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius about nine times that of Earth.

The sentiment is neutral  
50%



## 7. Click on **Next Step**

C aidemos.microsoft.com/text-analytics

## Text Analytics

- Step #1: Enter Your Message

Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius about nine times that of Earth.

[Next Step](#)

- Step #2: Sentiment & Key Phrases

This API analyzes your text to identify the keywords and discern the sentiment.

[Next Step](#)

- Step #3: Entity Linking

This API ascertains which of the key words are entities and links them in Wikipedia.

[Next Step](#)

- Step #4: Bing Entity Search

This API provides a summary of relevant information in the form of a card for each entity

[Start Over](#)

**Saturn**  
GENERIC  

Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius about nine times that of Earth...

[Read the rest on Wikipedia](#)

[See more on Bing](#)

**Saturn**  
GENERIC  

Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius of about nine...

[Read the rest on Wikipedia](#)

[See more on Bing](#)

**sixt**  
sixth  
ORGANIZATION  
<https://www.sixt.com/>  

Sixt SE is an international mobility service provider with about 2,100 locations in over 110 countries. Sixt SE acts as a parent and holding company...

[Read the rest on Wikipedia](#)

[See more on Bing](#)

**the Sun**  
FOOD  
<http://www.astronomy.ohio...>  

The Sun is the star at the center of the Solar System. It is a nearly perfect sphere of hot plasma, heated to incandescence by nuclear fusion reac...

[Read the rest on Wikipedia](#)

[See more on Bing](#)

**Sun**  
FOOD  
<http://www.astronomy.ohio...>  

The Sun is the star at the center of the Solar System. It is a nearly perfect sphere of hot plasma, heated to incandescence by nuclear fusion reac...

[Read the rest on Wikipedia](#)

[See more on Bing](#)

**second**  
SPECIALTY  

The second (symbol: s, abbreviation: sec) is the base unit of time in the International System of Units (SI) (French: Système International d'unité...  
(French: Système International d'unité...  
Read the rest on Wikipedia

[See more on Bing](#)

**Solar System**  
GENERIC  

The Solar System is the gravitationally bound system of the Sun and the objects that orbit it, either directly or indirectly. Of the objects that orbit th...

[Read the rest on Wikipedia](#)

[See more on Bing](#)

**Solar System**  
GENERIC  

The Solar System is the gravitationally bound system of the Sun and the objects that orbit it, either directly or indirectly. Of the objects that orbit th...

[Read the rest on Wikipedia](#)

[See more on Bing](#)

**Jupiter**  
GENERIC  

Jupiter is the fifth planet from the Sun and the largest in the Solar System. It is a gas giant with a mass one-thousandth that of the Sun, but two ...

[Read the rest on Wikipedia](#)

[See more on Bing](#)

**Jupiter**  

nine

[See more on Bing](#)

**Earth**  

[See more on Bing](#)

Now, After Text Analytics, Let's try another AI application "Language Understanding" where you can give commands in the format of text or voice and after understanding the command, it takes decisions accordingly. So, it's an application of Natural Language Processing. Let's get started

## 1.9 PRACTICAL : Language Understanding

1. Go to website <https://aidemos.microsoft.com/>
2. Select Language Understanding and click on "Try it out>" as shown in figure

aidemos.microsoft.com

**Machine Teaching**

The machine teaching deep reinforcement learning platform combines key innovations that simplify the development, training, and deployment of AI models giving you the tools to build subject matter expertise into complex industrial systems for your organization.

[Try it out >](#)



**Language Understanding**

Microsoft Cognitive Services Language Understanding interprets human language and understands the intent. Type in a custom intent or try one of the predefined ones to affect the house.

[Try it out >](#)

**AI Route Planner**

Bing Maps Distance Matrix API calculates travel time and distances for multi-stop routes and optimizes based on traffic patterns and other inputs. Try it to plan a sight-seeing trip.

[Try it out >](#)



**Video Indexer**

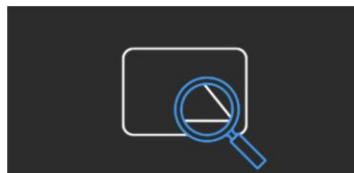
Video Indexer creates intelligent insights automatically after indexing a video, including people, keywords, sentiment, and more. Upload your own video today and gain new insights.

[Try it out >](#)

**Text Analytics**

Microsoft Cognitive Services Text Analytics API determines the sentiment of your message, typed or spoken. Try it out and see if the message is positive, negative, or neutral.

[Try it out >](#)



**Computer Vision**

Computer Vision can extract rich information from images to categorize and process visual data and perform machine-assisted moderation of images to help curate your services.

[Try it out >](#)

### 3. Click on See it on Action

aidemos.microsoft.com/luis

Meet the Language Understanding Service

A machine learning-based service to build natural language into apps, bots, and IoT devices. Quickly create enterprise-ready, custom models that continuously improve.



[See it in action >](#)

4. You can Give Whatever your Commands (Either by text or voice) and Switches will glow according in the house next to it.

Make a statement about lighting and LUIS will interpret and adjust the house accordingly

Make a statement (voice or text)

lights on Apply

Or select from one of the preset suggestions:

- Go to energy saver mode It's time to go to sleep
- I'm concentrating on my work I'm watching a movie
- Give me mood lighting for cooking It's too dark in here
- I need more light to read my book It's too dark to work in here
- I can't find the remote I need more light to cut carrots
- I'm expecting guests



## 5. Give Command Lights off and Click on Apply Button.

Make a statement about lighting and LUIS will interpret and adjust the house accordingly

Make a statement (voice or text)

lights off Apply

Or select from one of the preset suggestions:

- Go to energy saver mode It's time to go to sleep
- I'm concentrating on my work I'm watching a movie
- Give me mood lighting for cooking It's too dark in here
- I need more light to read my book It's too dark to work in here
- I can't find the remote I need more light to cut carrots
- I'm expecting guests Call Batman



## 6. Result

aidemos.microsoft.com/luis/demo

Microsoft | Microsoft AI Our approach Products & Services Blog Research All Microsoft Search Cart Sign in

Hands on with AI / Language Understanding

Make a statement about lighting and LUIS will interpret and adjust the house accordingly

Make a statement (voice or text)  **Apply**

Or select from one of the preset suggestions:

Go to energy saver mode | It's time to go to sleep  
I'm concentrating on my work | I'm watching a movie  
Give me mood lighting for cooking | It's too dark in here  
I need more light to read my book | It's too dark to work in here  
I can't find the remote | I need more light to cut carrots  
I'm expecting guests | Call Batman



## 7. Give Command **Call Batman**, and Click on **Apply**.

aidemos.microsoft.com/luis/demo

Microsoft | Microsoft AI Our approach Products & Services Blog Research All Microsoft Search Cart Sign in

Hands on with AI / Language Understanding

Make a statement about lighting and LUIS will interpret and adjust the house accordingly

Make a statement (voice or text)  **Apply**

Or select from one of the preset suggestions:

Go to energy saver mode | It's time to go to sleep  
I'm concentrating on my work | I'm watching a movie  
Give me mood lighting for cooking | It's too dark in here  
I need more light to read my book | It's too dark to work in here  
I can't find the remote | I need more light to cut carrots  
I'm expecting guests | Call Batman



## Result



Make a statement about lighting and LUIS will interpret and adjust the house accordingly

Make a statement (voice or text) ! Apply

Or select from one of the preset suggestions:

Go to energy saver mode It's time to go to sleep  
I'm concentrating on my work I'm watching a movie  
Give me mood lighting for cooking It's too dark in here  
I need more light to read my book It's too dark to work in here  
I can't find the remote I need more light to cut carrots  
I'm expecting guests Mission Accomplished

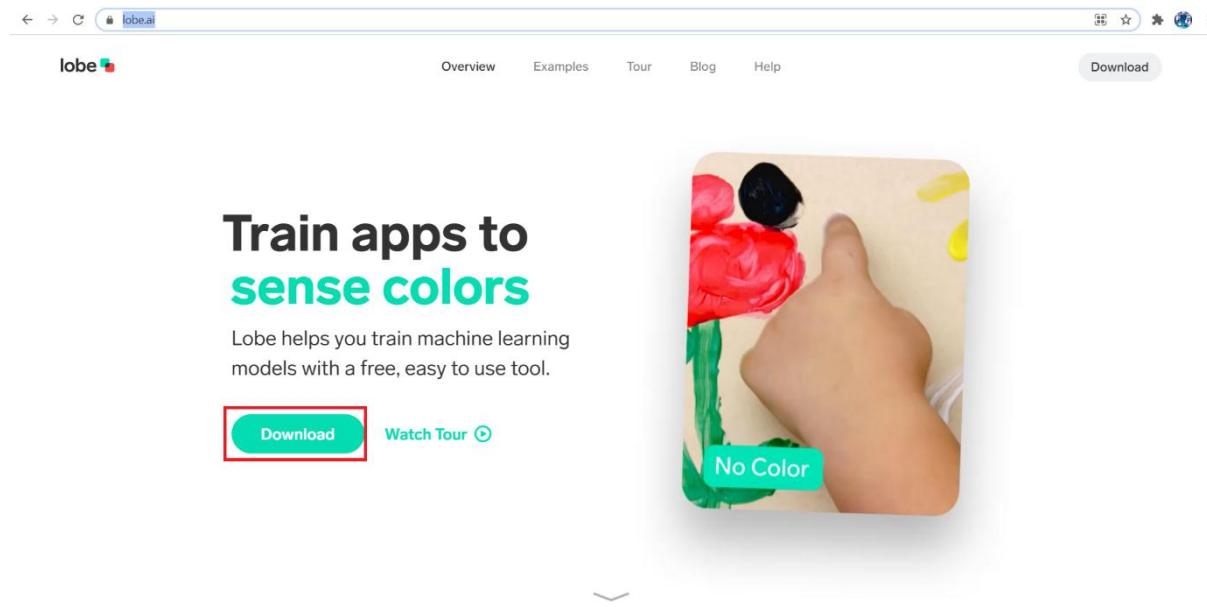
## What is a Chatbot?

A chatbot is a computer program that simulates human conversation through voice commands or text chats or both. Chatbot, short for chatterbot, is an artificial intelligence (AI) feature that can be embedded and used through any major messaging applications. There are a number of synonyms for chatbot, including "talk bot," "bot," "IM bot," "interactive agent" or "artificial conversation entity."

Now, we will try out an interesting machine learning application called lobe which can do a task of an image classification with few steps like labelling your images, training a model and understanding your results. Let's get started.

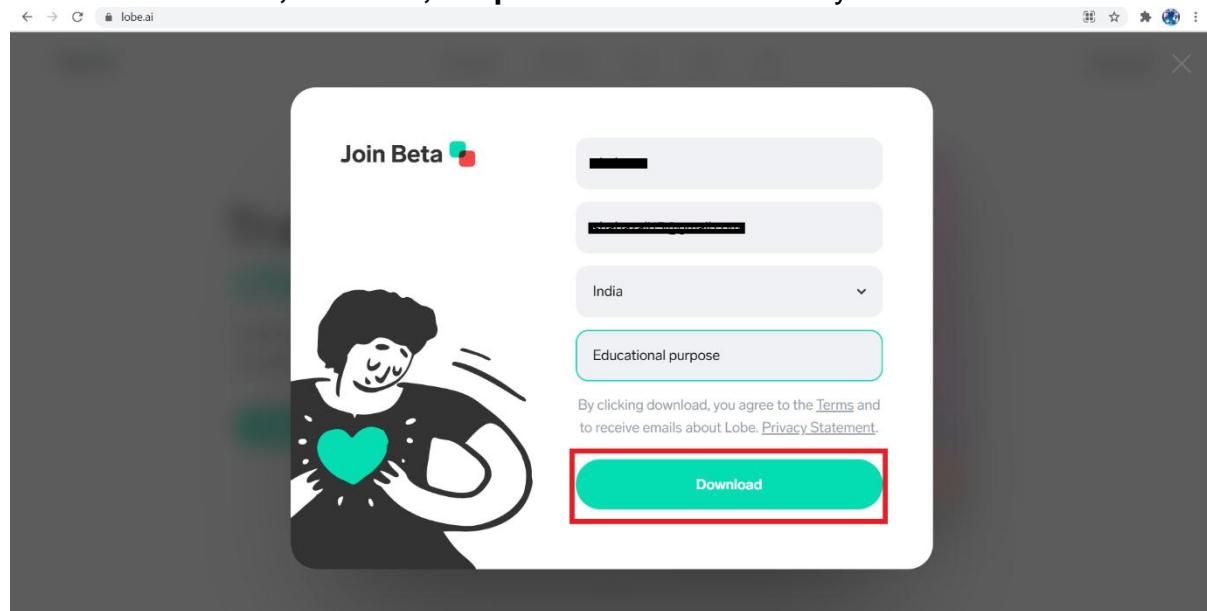
## 1.10 PRACTICAL : Image Recognition on Web Browser

1. Visit <https://lobe.ai/> Website
2. Click on Download



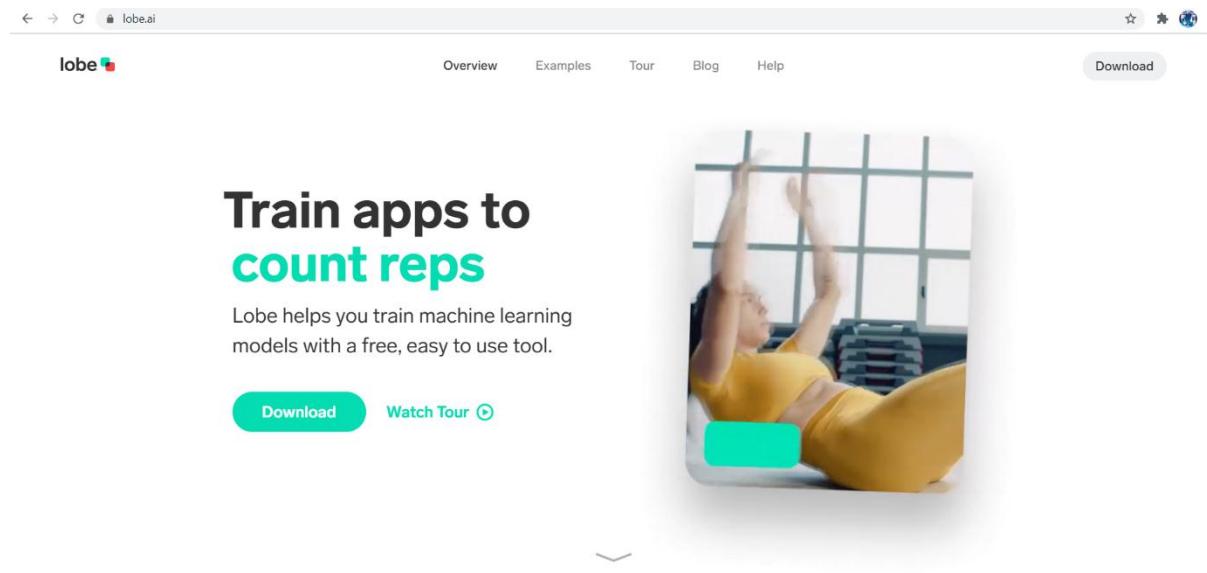
The screenshot shows the Lobe AI website. At the top, there's a navigation bar with links for Overview, Examples, Tour, Blog, Help, and Download. Below the navigation, the main heading reads "Train apps to sense colors". A subtext below it says "Lobe helps you train machine learning models with a free, easy to use tool." There are two buttons: a red-bordered "Download" button and a "Watch Tour" button. To the right, there's a small image of a hand interacting with a colorful flower, with a green button labeled "No Color" overlaid.

3. Enter Name, Email ID, Purpose. Select the Country and Click **Download**.



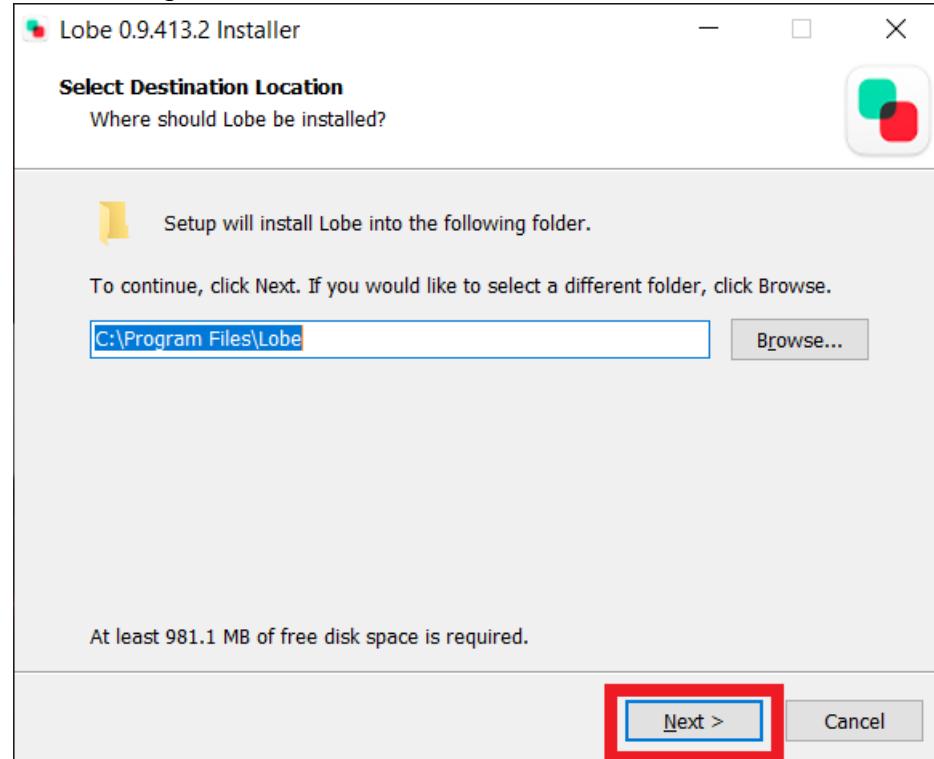
The screenshot shows a modal window titled "Join Beta". It contains fields for Name (two redacted lines), Email (redacted line), Country (set to India), and Purpose (set to "Educational purpose"). Below the form is a note about agreeing to Terms and Privacy Statement. At the bottom is a large red-bordered "Download" button.

4. File is being **Downloaded**

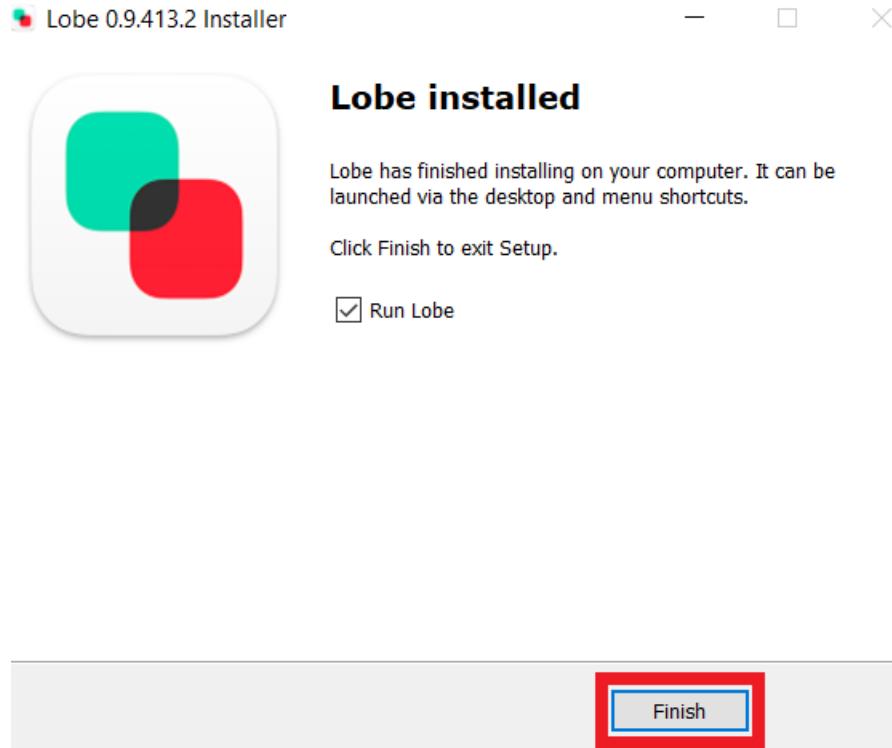


The screenshot shows the Lobe AI website. The main heading is "Train apps to count reps". Below it, a subtext says "Lobe helps you train machine learning models with a free, easy to use tool." There are two buttons: "Download" and "Watch Tour". To the right is a photograph of a person in yellow athletic wear performing sit-ups, with their arms raised. A green rectangular box highlights the "Download" button.

5. Start Installing the Downloaded file. And Click **NEXT**



6. Installation will take place. After that Click on **Finish**



7. Click on **Agreed**



## Welcome to Lobe

Lobe is subject to our [Software License](#). By using it, you are agreeing to the license, and our [Privacy Statement](#).

**Agreed**

8. Click on **Okay**



## Help us improve

Share crash reports and analytics to help us improve Lobe.

Your project images and models will never be shared.

No Thanks

Okay

9. Click on **Get Started**

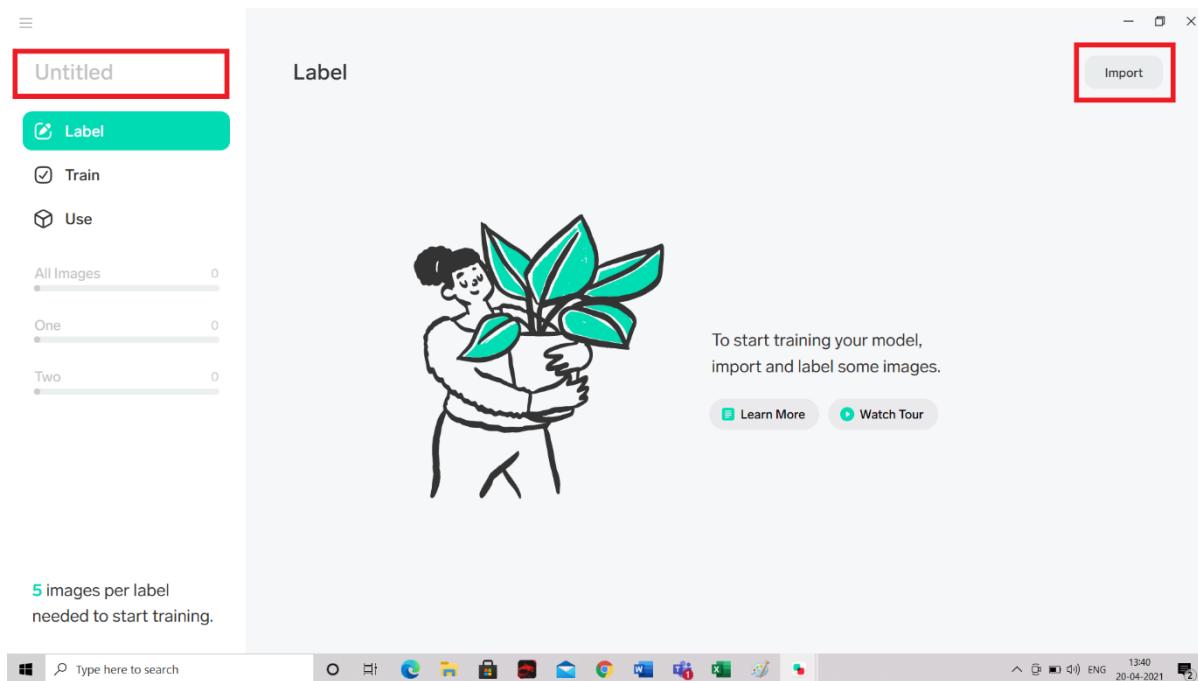


## You are all set!

Alright, you are ready to start training your first custom machine learning model with Lobe.

Get Started

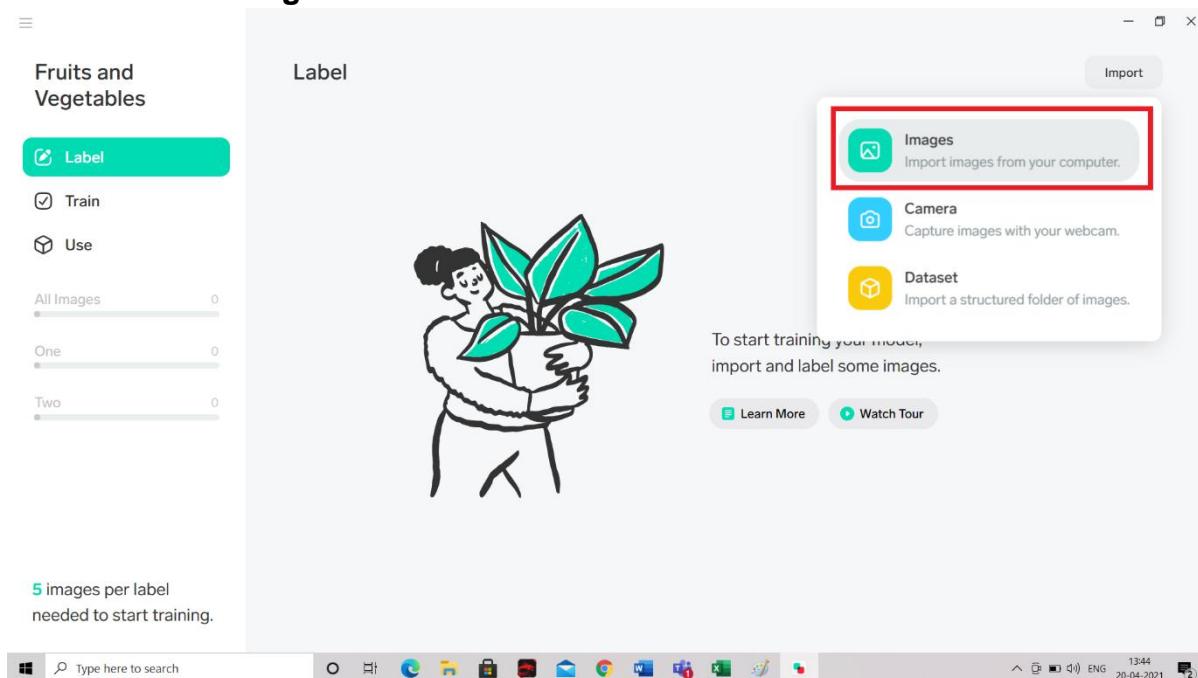
10. Name the Project. And Click on **Import**. To Upload Images from the System.  
(NOTE: My project is to train my system Difference between Fruits and Vegetables)



5 images per label  
needed to start training.



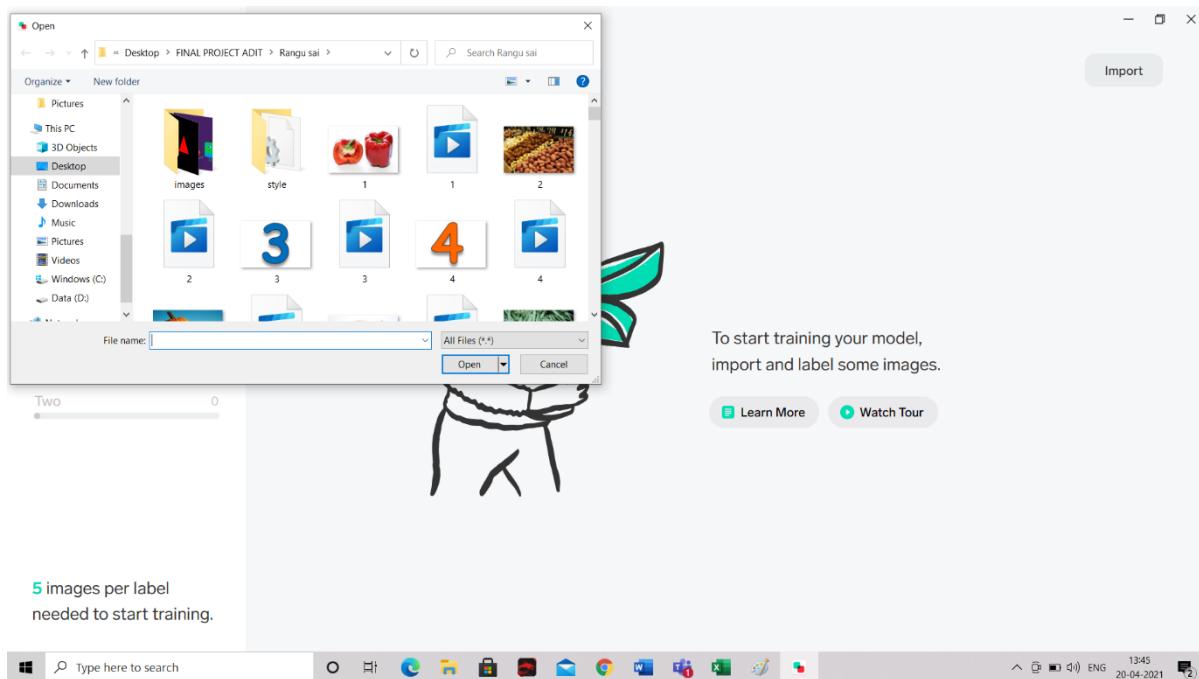
## 11. Click on Images



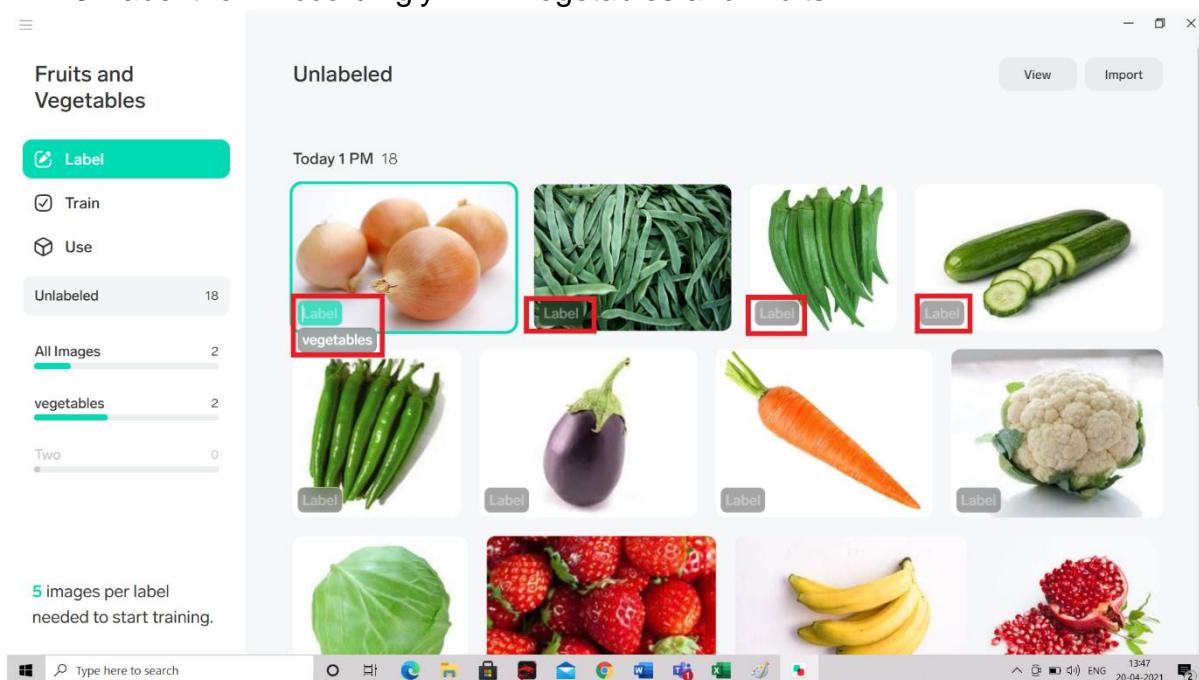
5 images per label  
needed to start training.



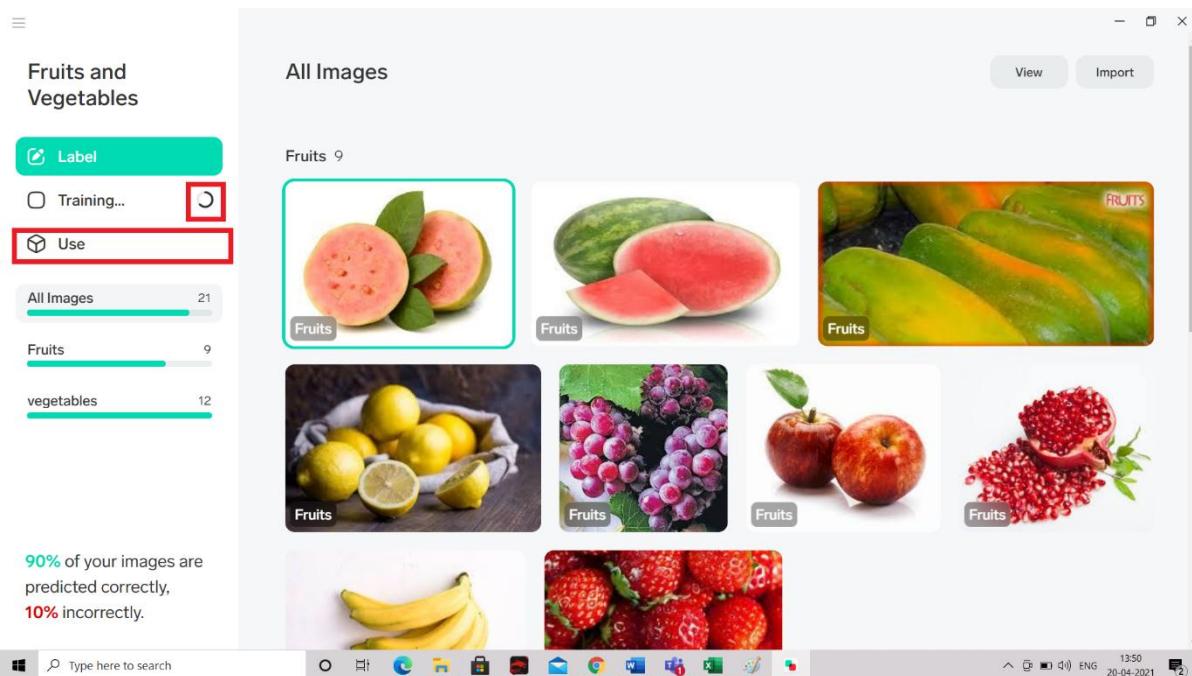
## 12. Select all you Project Images



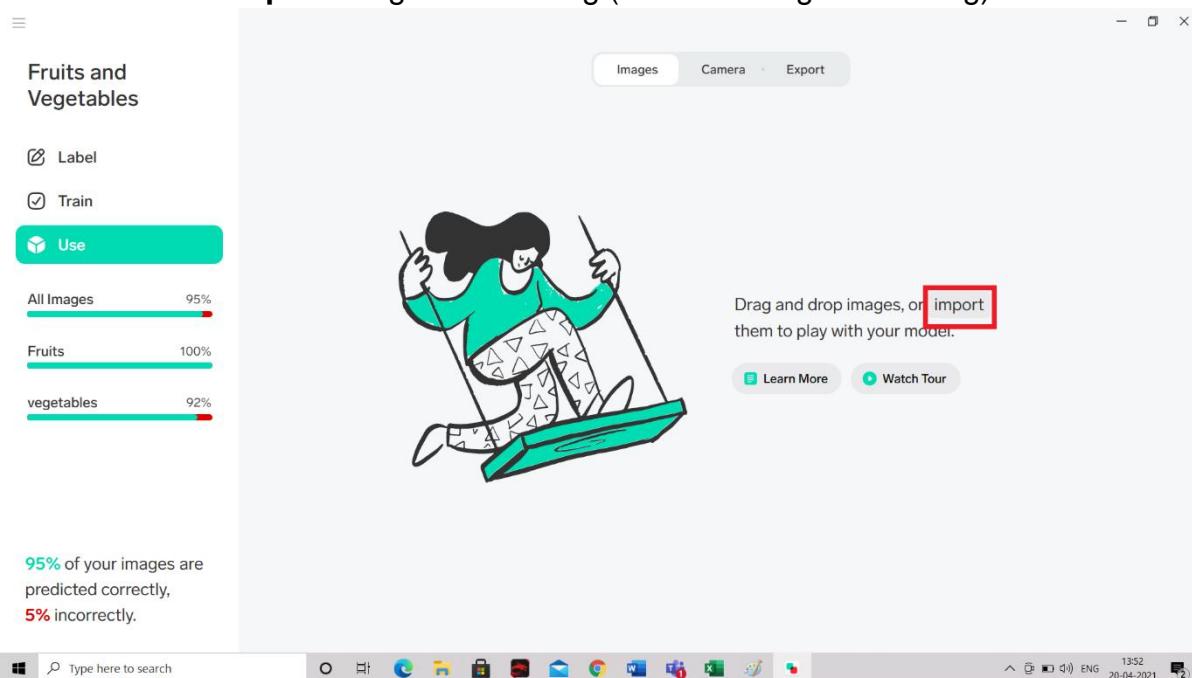
### 13. Label them Accordingly EX:- Vegetables and Fruits



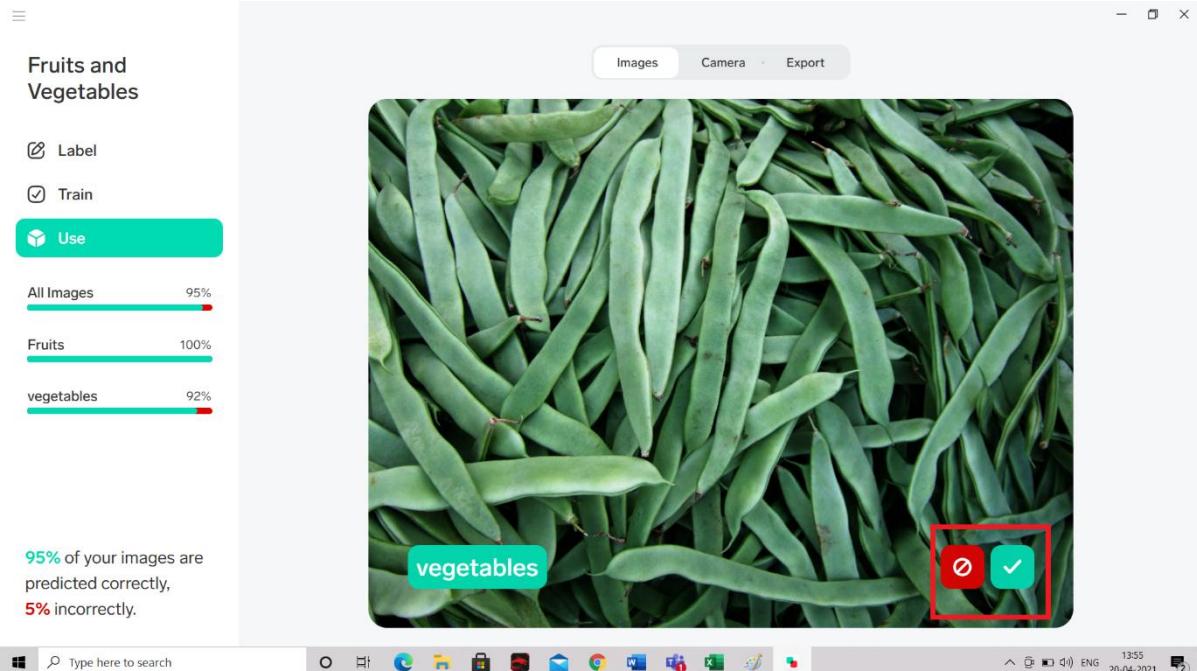
### 14. Training Will be done Automatically and Click on USE



### 15. Click on Import images for Testing (Select a image for testing)



16.The Result will be Shown Down. If it is Correct Click on Right. If it is wrong Click on Wrong Which will help to learn more to the system



So, in this chapter we have seen the history, market trends and opportunities in the very exciting field of Artificial Intelligence and the fundamental concepts in AI and machine learning. We also explored practically some of the applications of AI and Machine Learning like sentiment analysis, Natural Language Processing and Image classification. We explored an overall understanding of what is AI, ML and its real-life applications.

In the next chapter we will be discussing about basics of the Linux operating system, Data Analysis, Anaconda Software and key concepts in Python Programming language.

# Chapter 2 : Linux Kernel and Data Analysis with Python

## Learning Outcomes:

- Understand Linux Kernel, commands and functionalities
- Learn how to use Anaconda Software and Python Libraries
- Explore the fundamentals of Data Analysis
- Understand key concepts of Python Programming

## 2.1 Introduction to Linux

Linux is a Unix-like computer operating system assembled under the model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released 5 October 1991 by Linus Torvalds.

Linux was originally developed as a free operating system for Intel x86-based personal computers. It has since been ported to more computer hardware platforms than any other operating system. It is a leading operating system on servers and other big iron systems such as mainframe computers and supercomputers more than 90% of today's 500 fastest supercomputers run some variant of Linux, including the 10 fastest. Linux also runs on embedded systems (devices where the operating system is typically built into the firmware and highly tailored to the system) such as mobile phones, tablet computers, network routers, televisions and video game consoles; the Android system in wide use on mobile devices is built on the Linux kernel.

Linux is one of popular version of UNIX operating System. It is open source as its source code is freely available. It is free to use. Linux was designed considering UNIX compatibility. Its functionality list is quite similar to that of UNIX.

## Components of Linux System

Linux Operating System has primarily three components

- **Kernel** – Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
- **System Library** – System libraries are special functions or programs using which application programs or system utilities access Kernel's features. These libraries implement most of the functionalities of the operating system and do not require kernel module's code access rights.
- **System Utility** – System Utility programs are responsible to do specialized, individual level tasks.

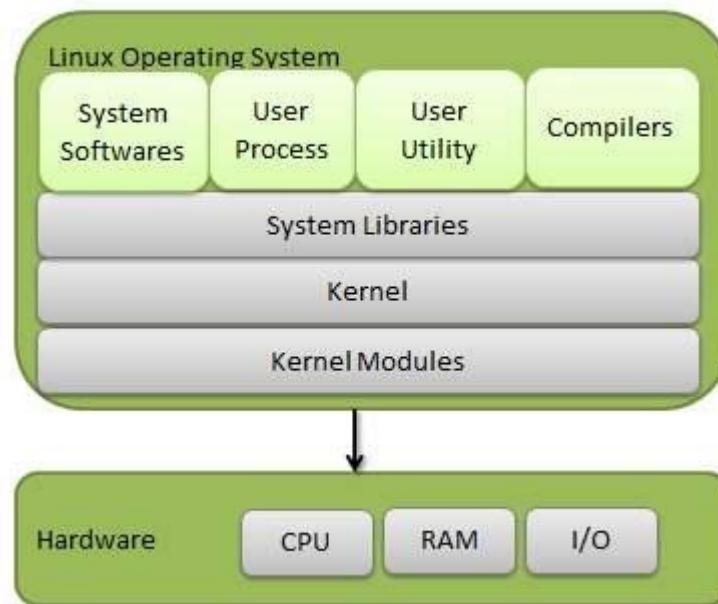


Image: LINUX Operating System [11]

Reference: [https://www.tutorialspoint.com/operating\\_system/os\\_linux.htm](https://www.tutorialspoint.com/operating_system/os_linux.htm)

## Kernel Mode vs User Mode

Kernel component code executes in a special privileged mode called **kernel mode** with full access to all resources of the computer. This code represents a single process, executes in single address space and does not require any context switch and hence is very efficient and fast. Kernel runs each processes and provides system services to processes, provides protected access to hardware to processes.

Support code which is not required to run in kernel mode is in System Library. User programs and other system programs work in **User Mode** which has no access to system hardware and kernel code. User programs/ utilities use System libraries to access Kernel functions to get system's low level tasks.

## Basic Features

Following are some of the important features of Linux Operating System.

**Portable** - Portability means software can work on different types of hardware in same way. Linux kernel and application programs support their installation on any kind of hardware platform.

**Open Source** - Linux source code is freely available, and it is community-based development project. Multiple Teams works in collaboration to enhance the capability of Linux operating system and it is continuously evolving.

**Multi-User** - Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.

**Multiprogramming** - Linux is a multiprogramming system means multiple applications can run at same time.

**Hierarchical File System** - Linux provides a standard file structure in which system files/ user files are arranged.

**Shell** - Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs etc.

**Security** - Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

## Linux Advantages

1. **Low cost:** You don't need to spend time and money to obtain licenses since Linux and much of its software come with the GNU General Public License. You can start to work immediately without worrying that your software may stop working anytime because the free trial version expires. Additionally, there are large repositories from which you can freely download high quality software for almost any task you can think of.

2. **Stability:** Linux doesn't need to be rebooted periodically to maintain performance levels. It doesn't freeze up or slow down over time due to memory leaks and such. Continuous up-times of hundreds of days (up to a year or more) are not uncommon.

3. **Performance:** Linux provides persistent high performance on workstations and on networks. It can handle unusually large numbers of users simultaneously and can make old computers sufficiently responsive to be useful again.

4. **Network friendliness:** Linux was developed by a group of programmers over the Internet and has therefore strong support for network functionality; client and server systems can be easily set up on any computer running Linux. It can perform tasks such as network backups faster and more reliably than alternative systems.

5. **Flexibility:** Linux can be used for high performance server applications, desktop applications, and embedded systems. You can save disk space by only installing the components needed for a particular use. You can restrict the use of specific computers by installing for example only selected office applications instead of the whole suite.

6. **Compatibility:** It runs all common UNIX software packages and can process all common file formats.

7. **Choice:** The large number of Linux distributions gives you a choice. Each distribution is developed and supported by a different organization. You can pick the one you like best; the core functionalities are the same; most software runs on most distributions.
8. **Fast and easy installation:** Most Linux distributions come with user-friendly installation and setup programs. Popular Linux distributions come with tools that make installation of additional software very user friendly as well.
9. **Full use of hard disk:** Linux continues work well even when the hard disk is almost full.
10. **Multi-tasking:** Linux is designed to do many things at the same time; e.g., a large printing job in the background won't slow down your other work.
11. **Security:** Linux is one of the most secure operating systems. —Walls!! and flexible file access permission systems prevent access by unwanted visitors or viruses. Linux users have the option to select and safely download software, free of charge, from online repositories containing thousands of high-quality packages. No purchase transactions requiring credit card numbers or other sensitive personal information are necessary.
12. **Open Source:** If you develop software that requires knowledge or modification of the operating system code, LINUX's source code is at your fingertips. Most Linux applications are Open Source as well.

## Linux Distribution (Operating System) Names

A few popular names:

1. Redhat Enterprise Linux
2. Fedora Linux
3. Debian Linux
4. Suse Enterprise Linux
5. Ubuntu Linux

### *Common things between Linux & UNIX*

Both share many common applications such as:

1. GUI, file, and windows managers (KDE, Gnome)
2. Shells (ksh, csh, bash)
3. Various office applications such as OpenOffice.org
4. Development tools (perl, php, python, GNU c/c++ compilers)
5. Posix interface

### *Layered Architecture:*

Linux System Architecture is consisting of following layers

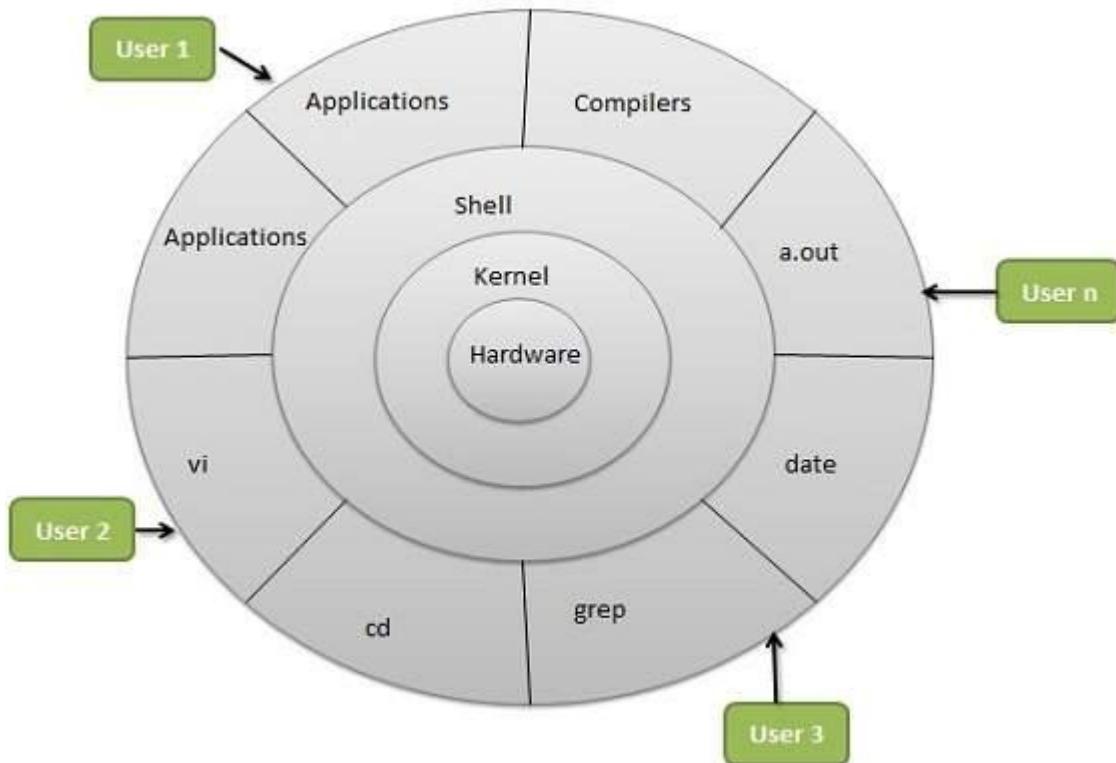


Image: Basic Block diagram of Layered Architecture of LINUX  
 Reference: [https://www.tutorialspoint.com/operating\\_system/os\\_linux.html](https://www.tutorialspoint.com/operating_system/os_linux.html)

**Hardware layer** - Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).

**Kernel** - Core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.

**Shell** - An interface to kernel, hiding complexity of kernel's functions from users. Takes commands from user and executes kernel's functions.

**Utilities** - Utility programs giving user most of the functionalities of an operating systems.

### LINUX File system

Linux file structure files are grouped according to purpose. Ex: commands, data files, documentation. Parts of a Unix directory tree are listed below. All directories are grouped under the root entry "/". That part of the directory tree is left out of the below diagram.

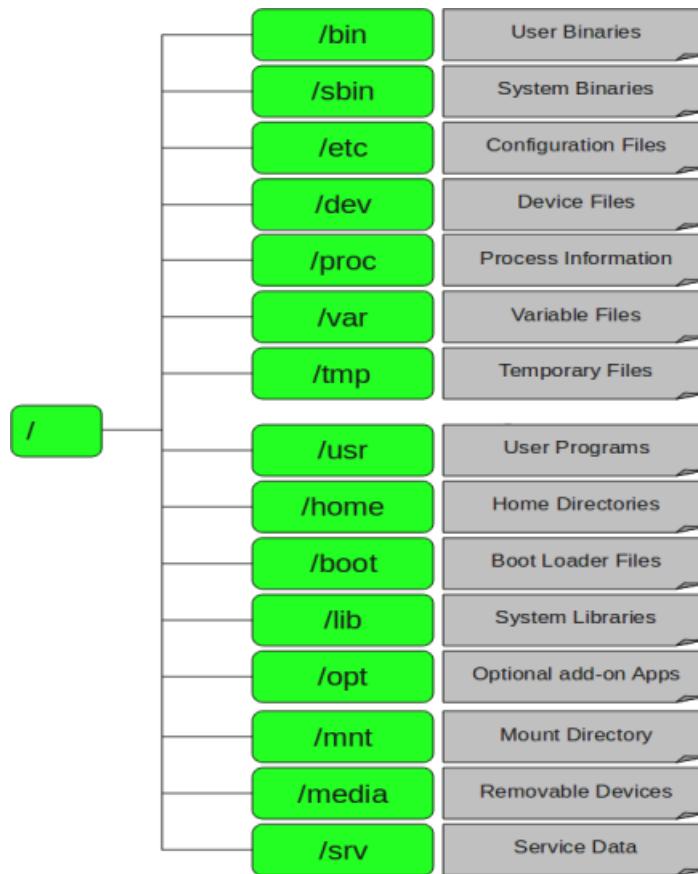


Image: LINUX File Structure

Reference: [https://www.tutorialspoint.com/operating\\_system/os\\_linux.html](https://www.tutorialspoint.com/operating_system/os_linux.html)

## 1. / – Root

- Every single file and directory starts from the root directory.
- Only root user has write privilege under this directory.
- Please note that /root is root user's home directory, which is not same as /.

## 2. /bin – User Binaries

- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here.
- For example: ps, ls, ping, grep, cp.

## 3. /sbin – System Binaries

- Just like /bin, /sbin also contains binary executables.
- But, the linux commands located under this directory are used typically by system administrator, for system maintenance purpose.
- For example: iptables, reboot, fdisk, ifconfig, swapon

#### **4. /etc – Configuration Files**

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- For example: /etc/resolv.conf, /etc/logrotate.conf

#### **5. /dev – Device Files**

- Contains device files.
- These include terminal devices, usb, or any device attached to the system.
- For example: /dev/tty1, /dev/usbmon0

#### **6. /proc – Process Information**

- Contains information about system process.
- This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example:  
/proc/uptime

#### **7. /var – Variable Files**

- var stands for variable files.
- Content of the files that are expected to grow can be found under this directory.
- This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

#### **8. /tmp – Temporary Files**

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.

#### **9. /usr – User Programs**

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under
- /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- /usr/lib contains libraries for /usr/bin and /usr/sbin

- /usr/local contains users programs that you install from source. For example, when you
- install apache from source, it goes under /usr/local/apache2

#### **10. /home – Home Directories**

- Home directories for all users to store their personal files.
- For example: /home/john, /home/nikita

#### **11. /boot – Boot Loader Files**

- Contains boot loader related files.
- Kernel initrd, vmlinuz, grub files are located under /boot
- For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

#### **12. /lib – System Libraries**

- Contains library files that supports the binaries located under /bin and /sbin
- Library filenames are either lib\* or lib\*.so.\*
- For example: lib-2.11.1.so, libncurses.so.5.7

#### **13. /opt – Optional add-on Applications**

- opt stands for optional.
- Contains add-on applications from individual vendors.
- add-on applications should be installed under either /opt/ or /opt/ sub-directory.

#### **14. /mnt – Mount Directory**

- Temporary mount directory where sysadmins can mount filesystems.

#### **15. /media – Removable Media Devices**

- Temporary mount directory for removable devices.
- For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives;
- /media/cdrecorder for CD writer

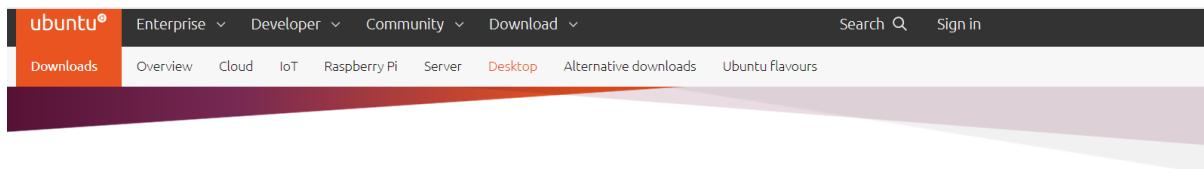
#### **16. /srv – Service Data**

- srv stands for service.
- Contains server specific services related data.
- For example, /srv/cvs contains CVS related data.

## **2.2 Installation of Ubuntu**

### **Downloading**

Visit this link to <http://www.ubuntu.com/download/desktop> download Ubuntu.



The screenshot shows the Ubuntu website's desktop download page. At the top, there's a navigation bar with links for Enterprise, Developer, Community, Download, Search, and Sign in. Below the navigation is a menu bar with Downloads, Overview, Cloud, IoT, Raspberry Pi, Server, Desktop (which is highlighted in orange), Alternative downloads, and Ubuntu Flavours. The main content area features a large image of a laptop with the Ubuntu desktop environment. Below the image, the text "Download Ubuntu Desktop" is prominently displayed. Underneath, it says "Ubuntu 20.04.2.0 LTS". A paragraph describes the LTS version, mentioning long-term support until April 2025. A red box highlights the green "Download" button. To the right, there's a link to alternative versions and mirrors.

## Download Ubuntu Desktop

### Ubuntu 20.04.2.0 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2025, of free security and maintenance updates, guaranteed.

[Ubuntu 20.04 LTS release notes](#)

Recommended system requirements:

- 2 GHz dual core processor or better
- Internet access is helpful
- 4 GB system memory
- Either a DVD drive or a USB port for the installer media
- 25 GB of free hard drive space

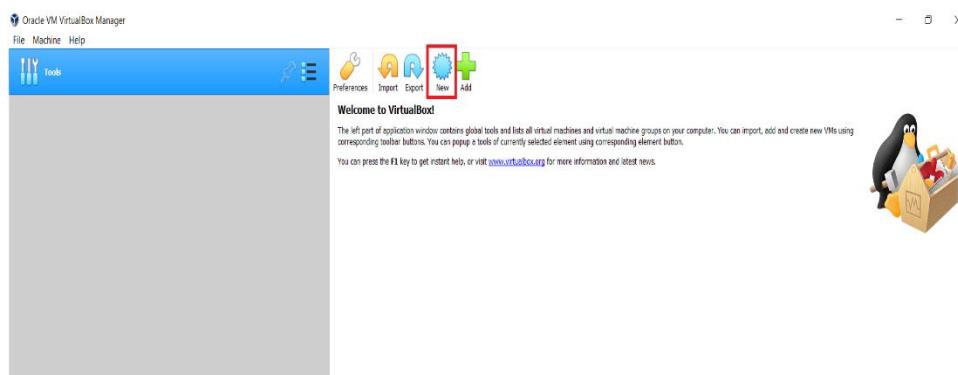
[Download](#)

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and past releases see our [alternative downloads](#).

You can select 32/64-bit versions as per your choice.

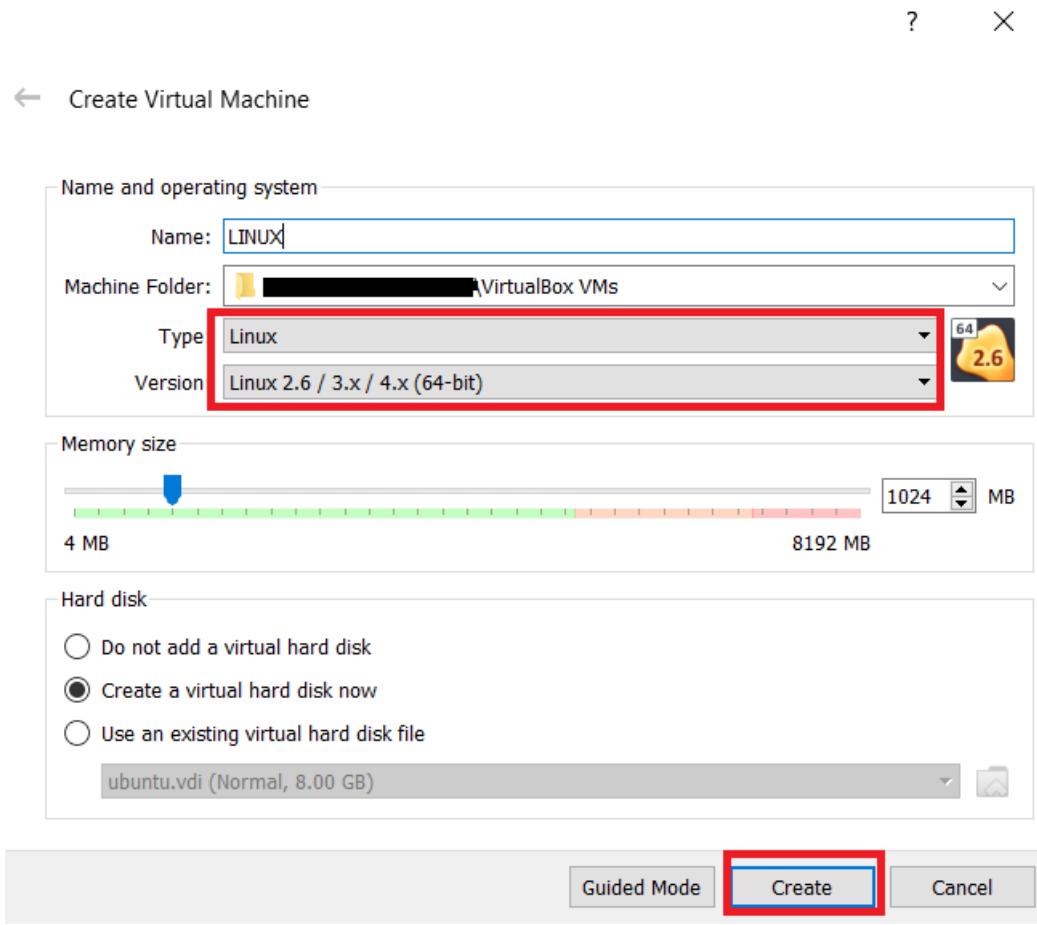
*Create a Machine in Virtual Box*

**Step-1)** Open Virtual box and click on new button



**Step-2)** In next window, give the name of your OS which you are installing in virtual box. And select OS like Linux and version as Ubuntu 32 bit. And click on next

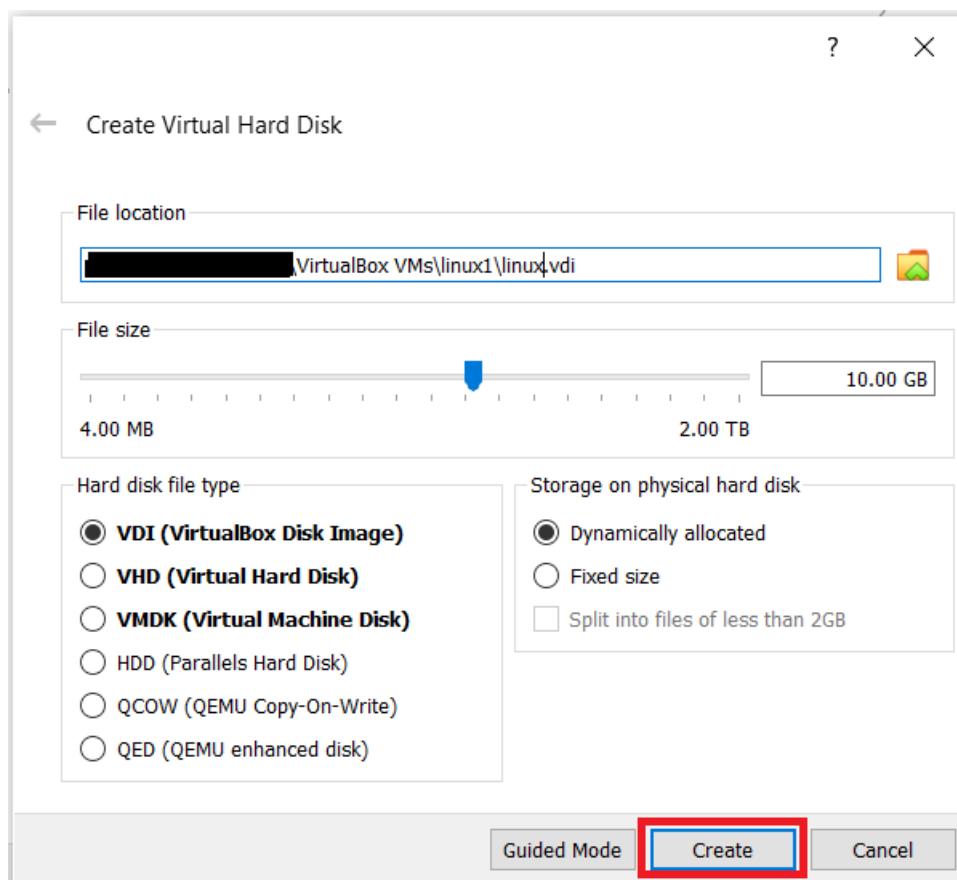
**Step-3)** Now Allocate Ram Size To your Virtual OS. I recommended keeping 1024mb (1 GB) ram to run Ubuntu better. And click on next.



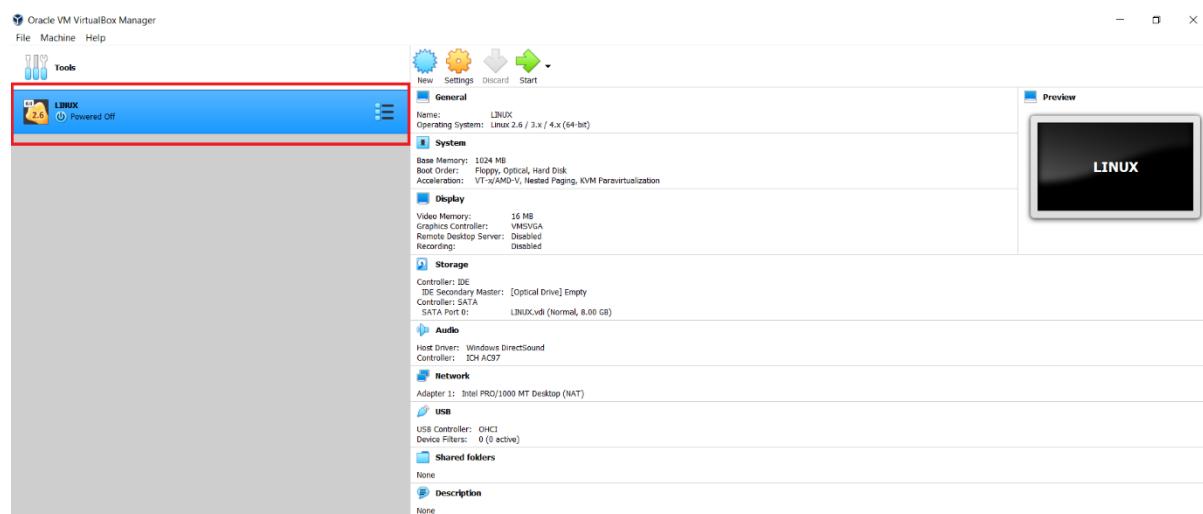
**Step-4)** Allocate memory to your virtual hard drive .10 GB recommended.

**Step-5)** select VDI (virtual hard disk) option

**Step-6)** Click on dynamic allocated. This means that the size of the disk will increase dynamically as per requirement. and click on create button.



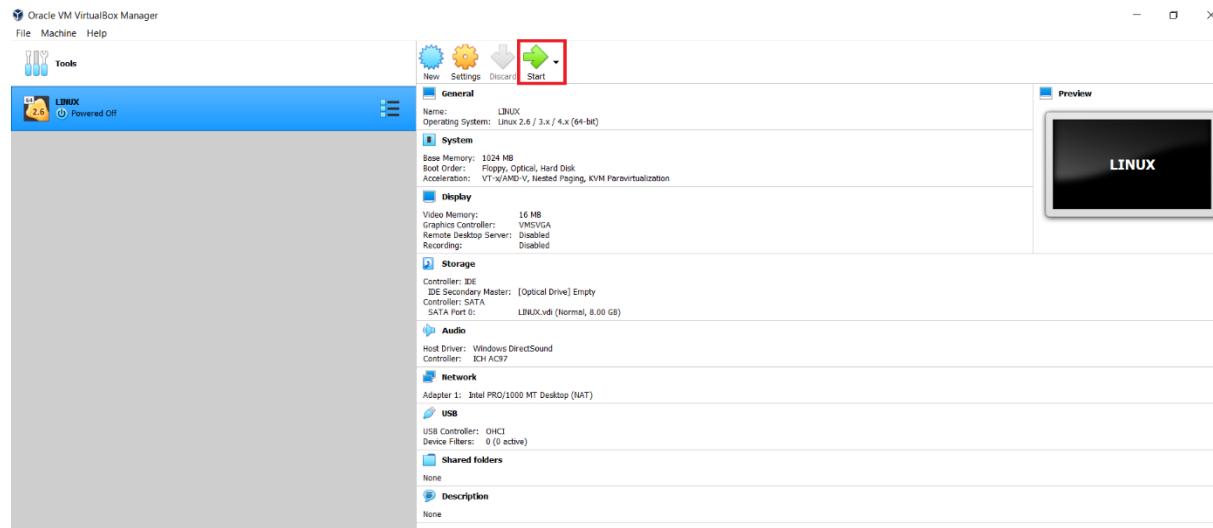
### Step-8) Now you can see the machine name in left panel



So a Machine (PC) with 8GB Hardisk, 1GB RAM is ready.

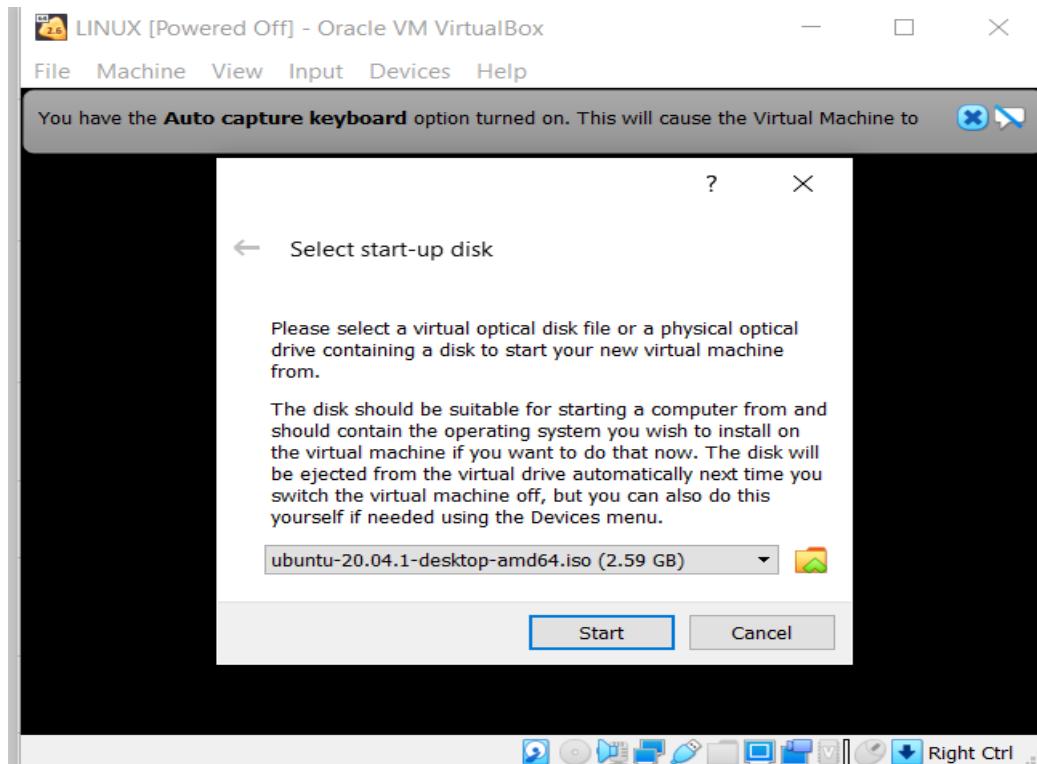
## How to Install Ubuntu

### Step 1) Select the Machine and Click on Start

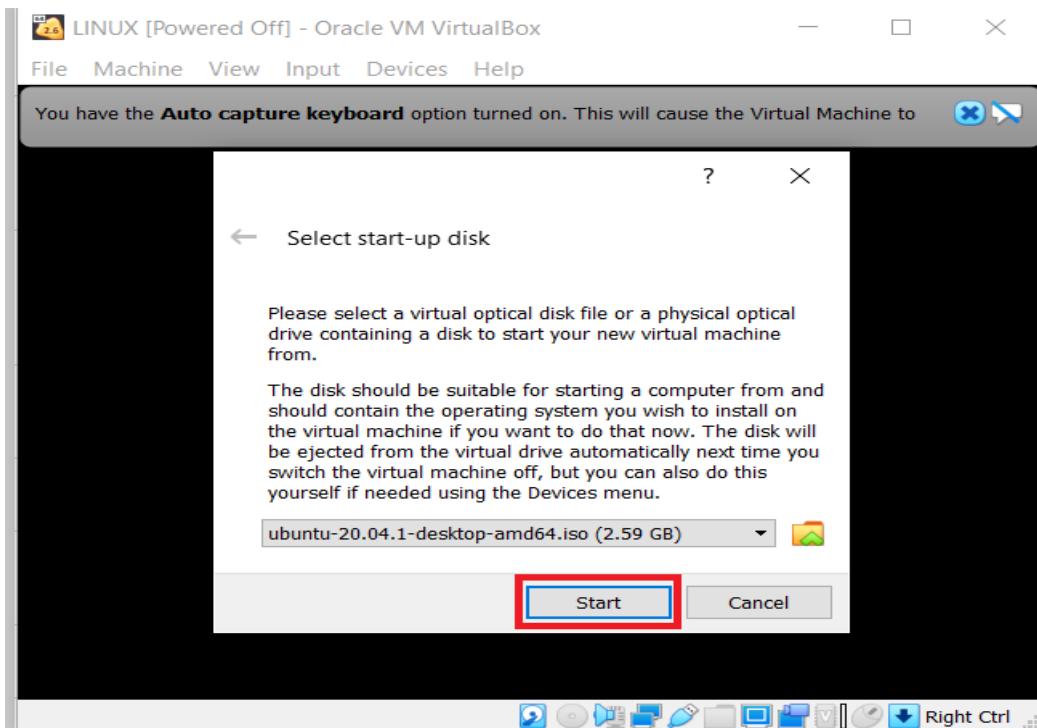


### Step 2) Select the Folder Option

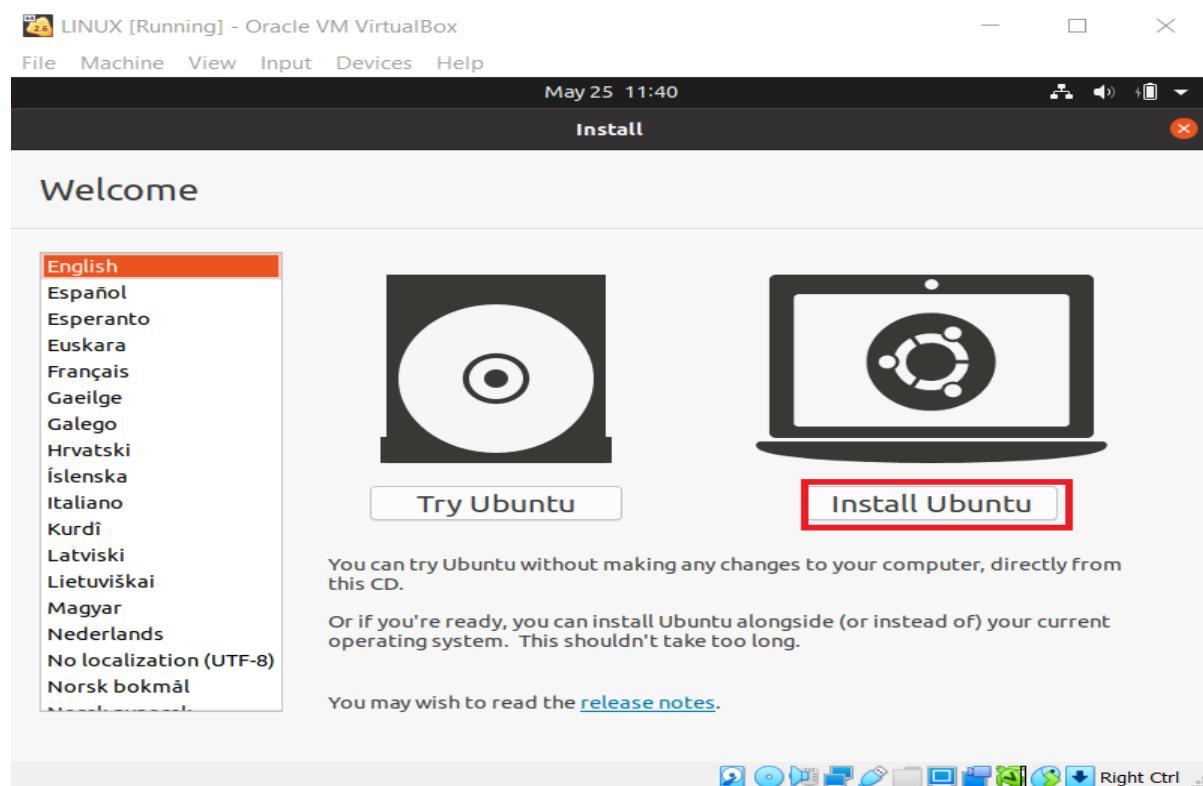
### Step 3) Select the Ubuntu iso file



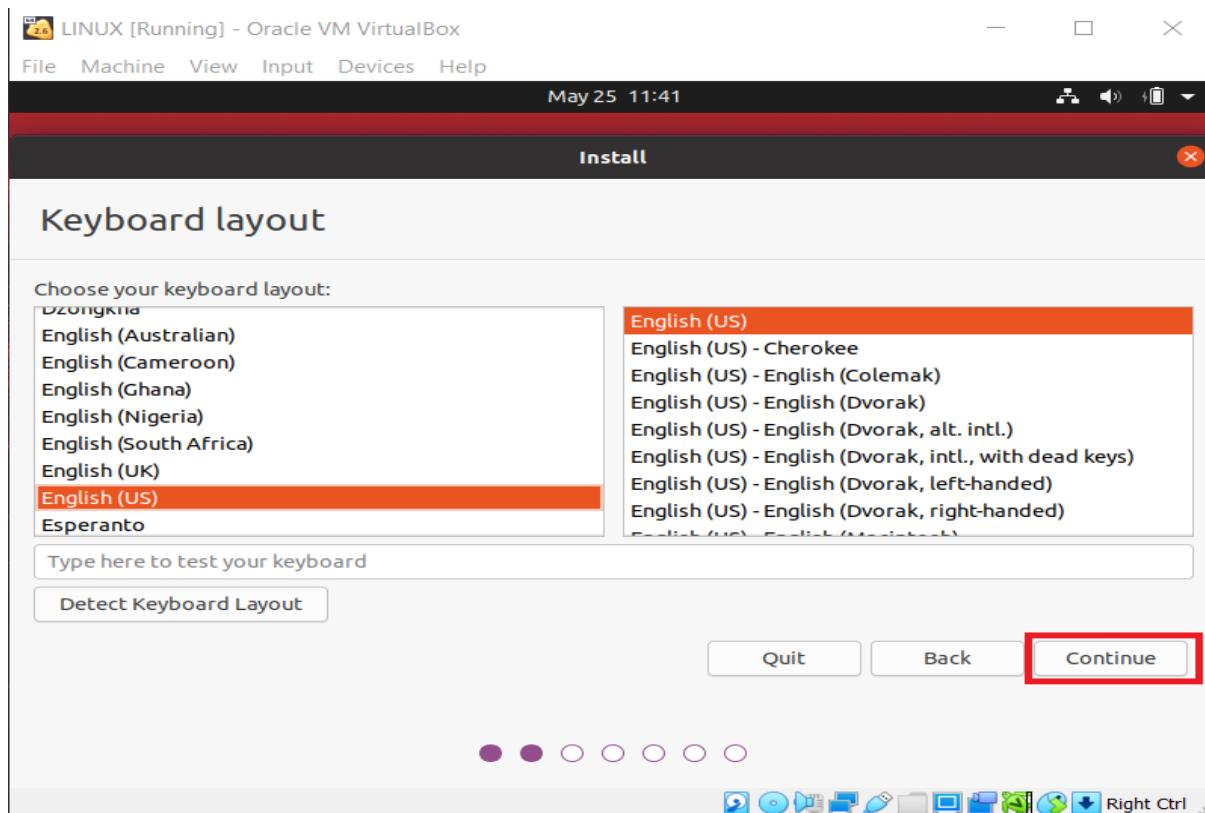
### Step 4) Click Start



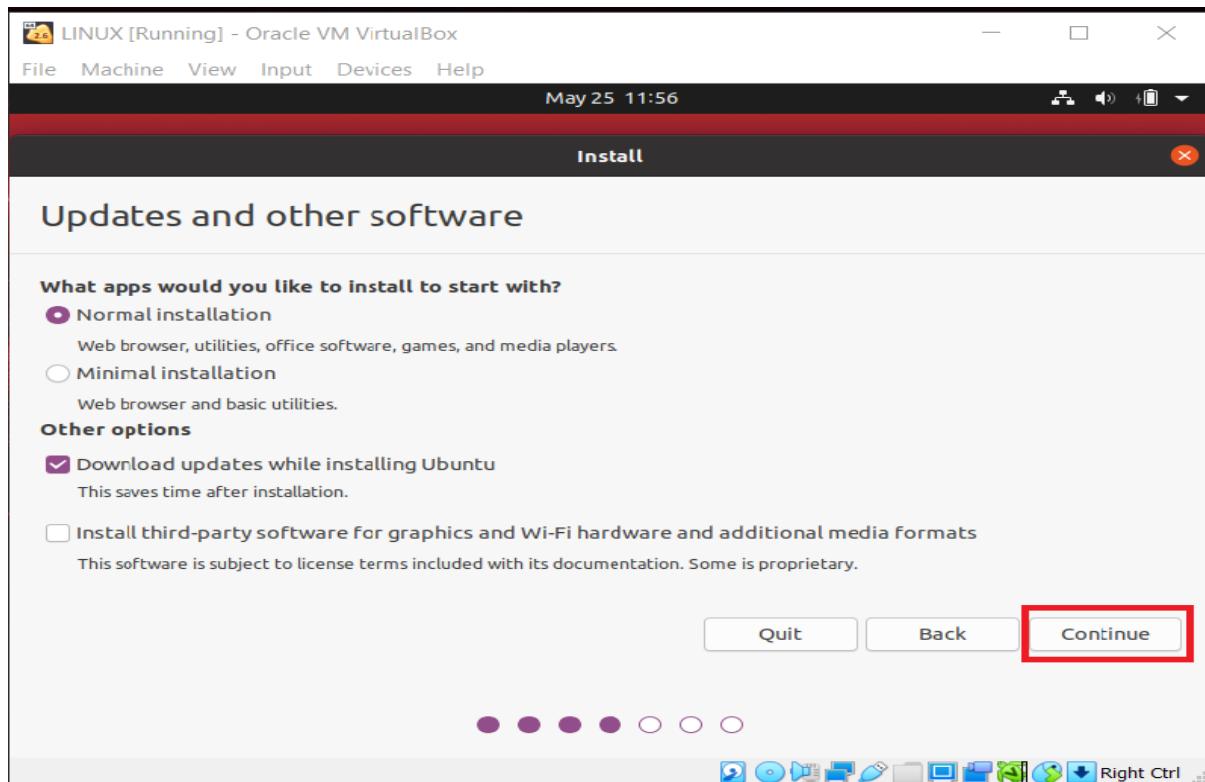
**Step-5)** You have an option to Run Ubuntu WITHOUT installing. In this tutorial will install Ubuntu



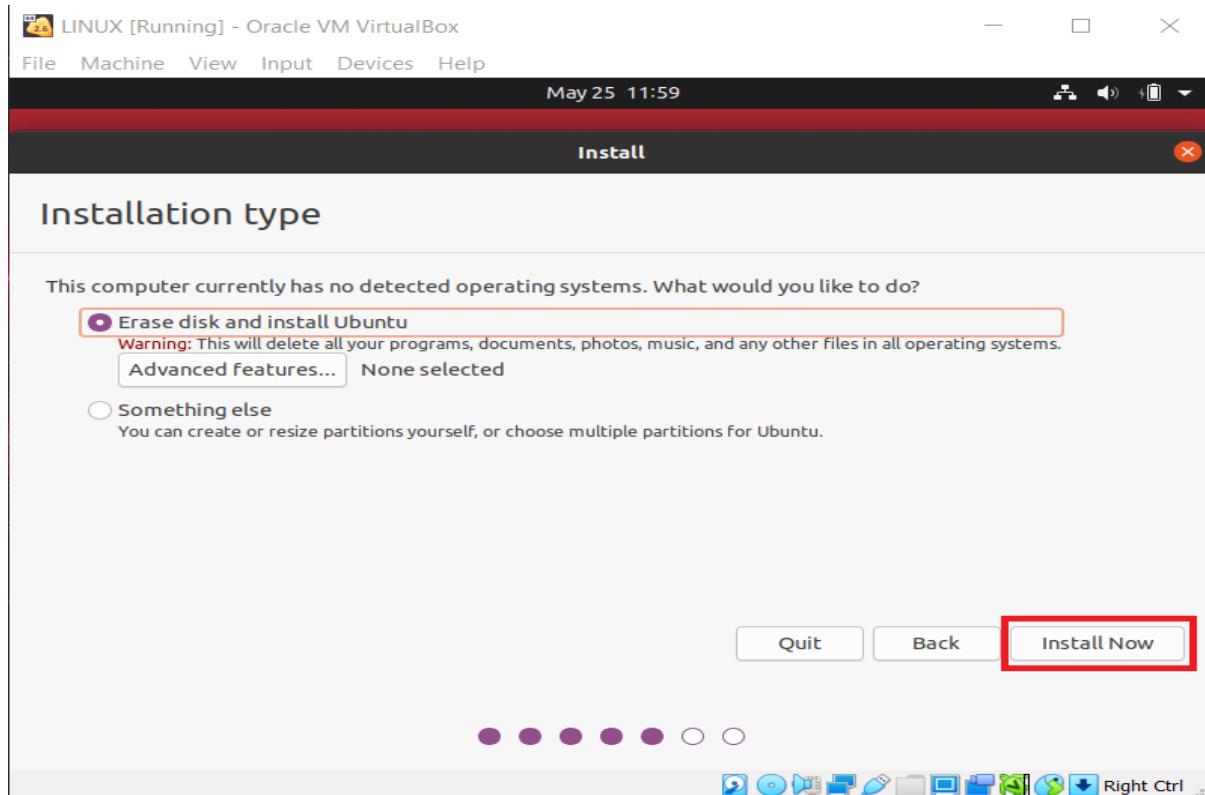
**Step-6)** Select your keyboard layout, by default English (US) is selected but if you want to change then, you can select in the list. And click on continue



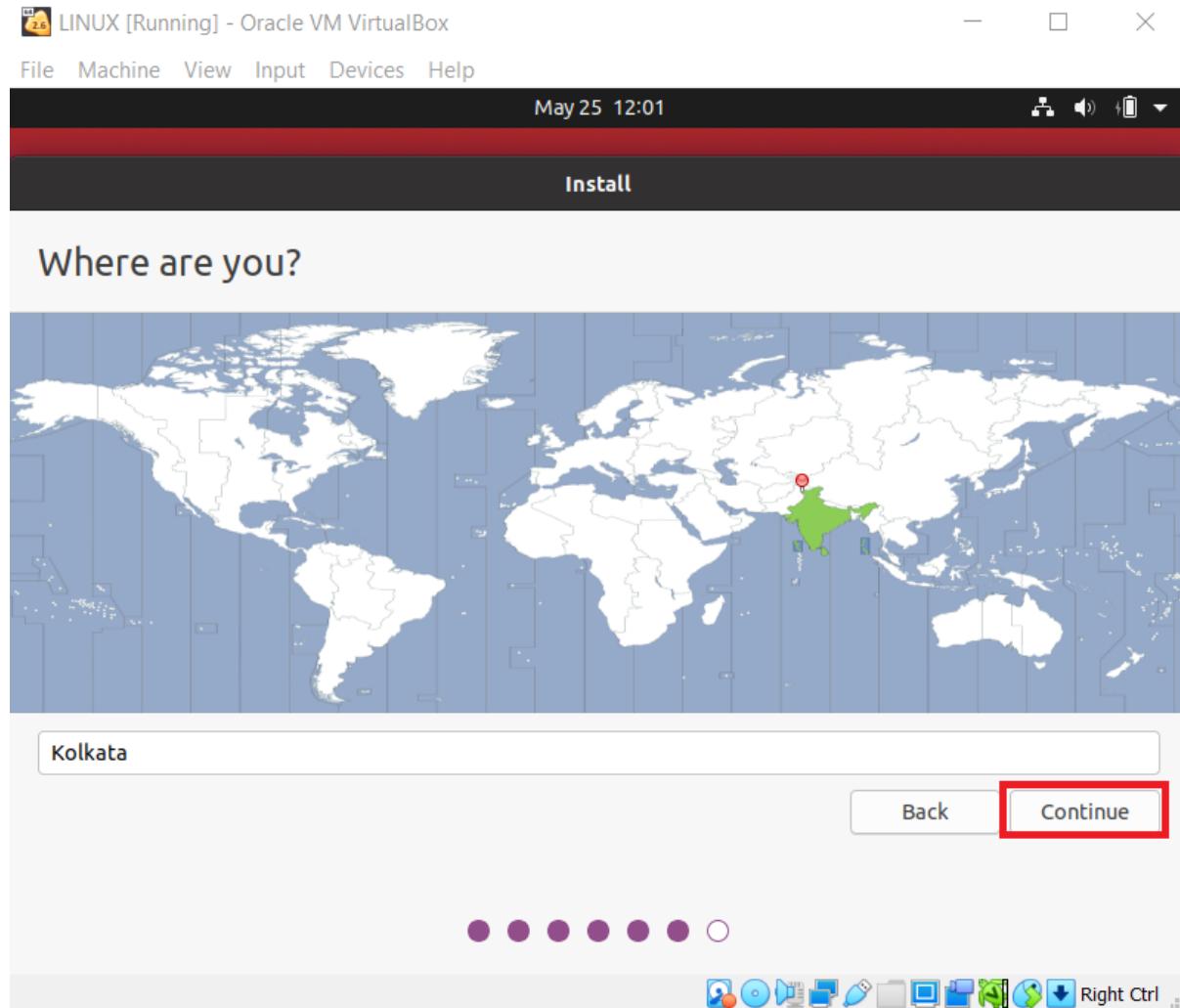
**Step-7) Click continue.**



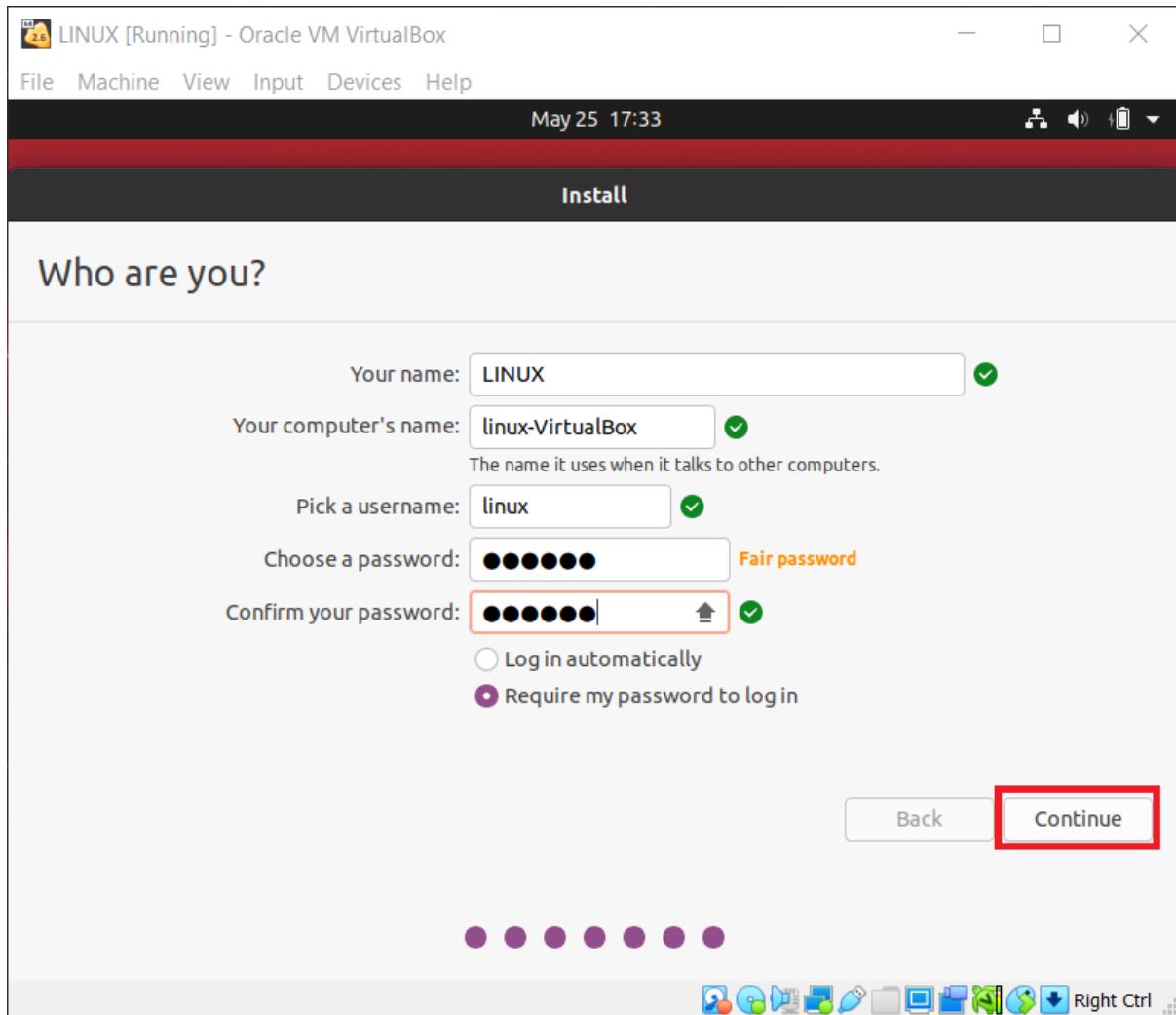
**Step-8)** Select option to erase the disk and install Ubuntu and click on install now. This option installs Ubuntu into our virtual hard drive which is we made earlier. It will not harm your PC or Windows installation



**Step-9)** Select your location for setting up time zone, and click on continue

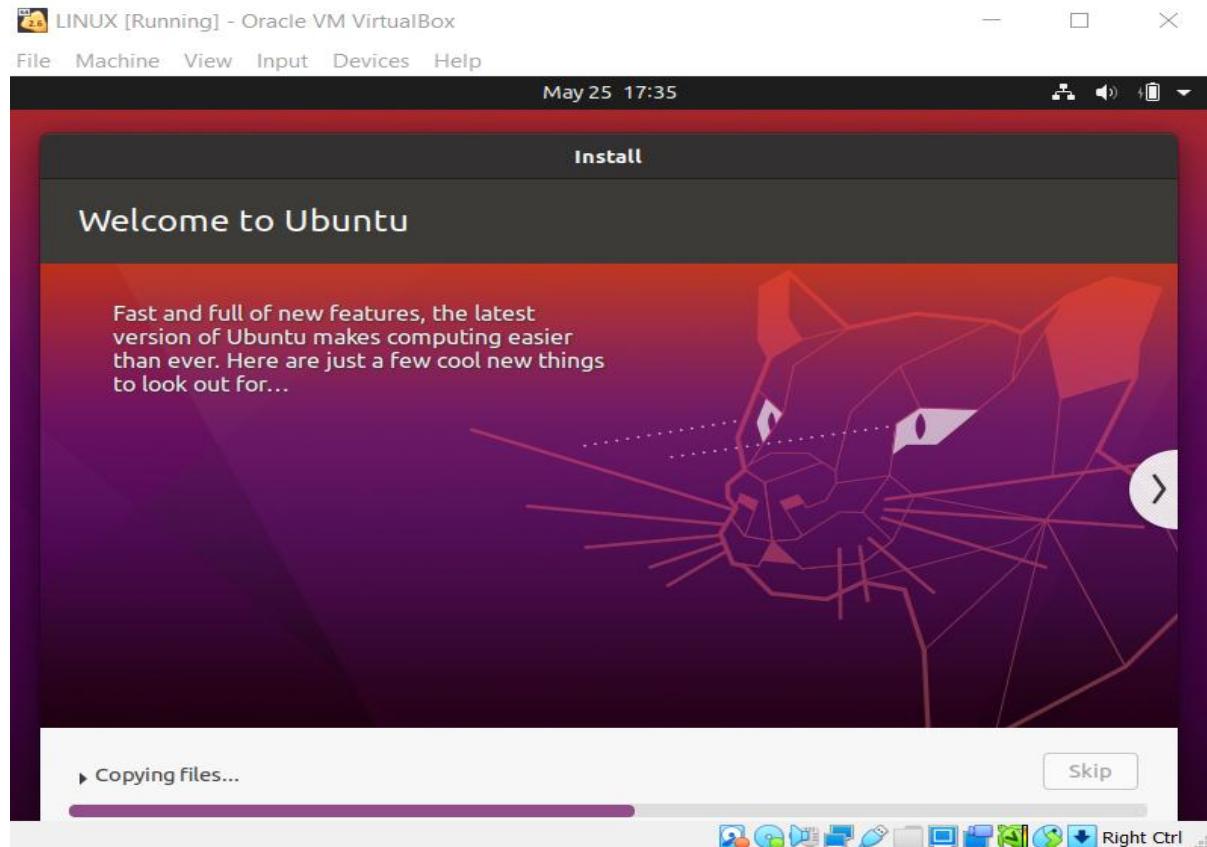


**Step-10)** Select your username and password for your Ubuntu admin account. This information has been needed for installing any software package into Ubuntu and also for login to your OS. Fill up your details and tick on login automatically to ignore

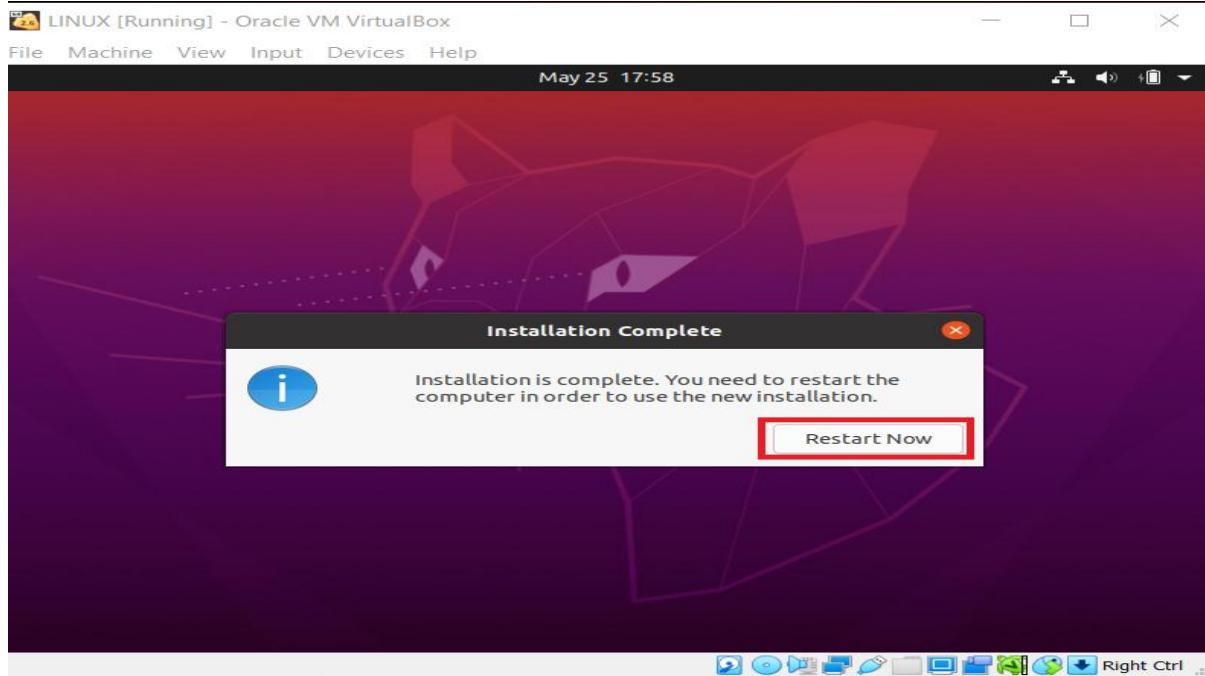


login attempt and click on continue

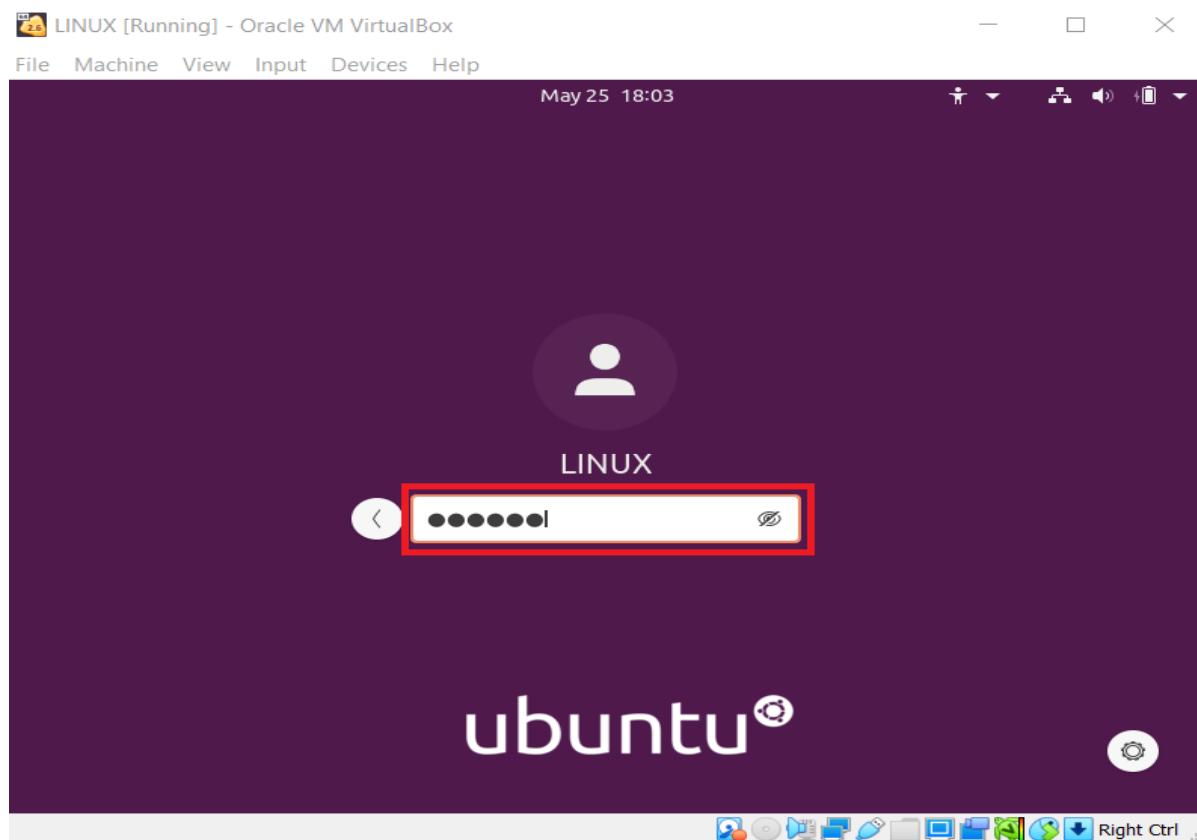
**Step-11)** Installation process starts. May take up to 30 minutes. Please wait until installation process completes.



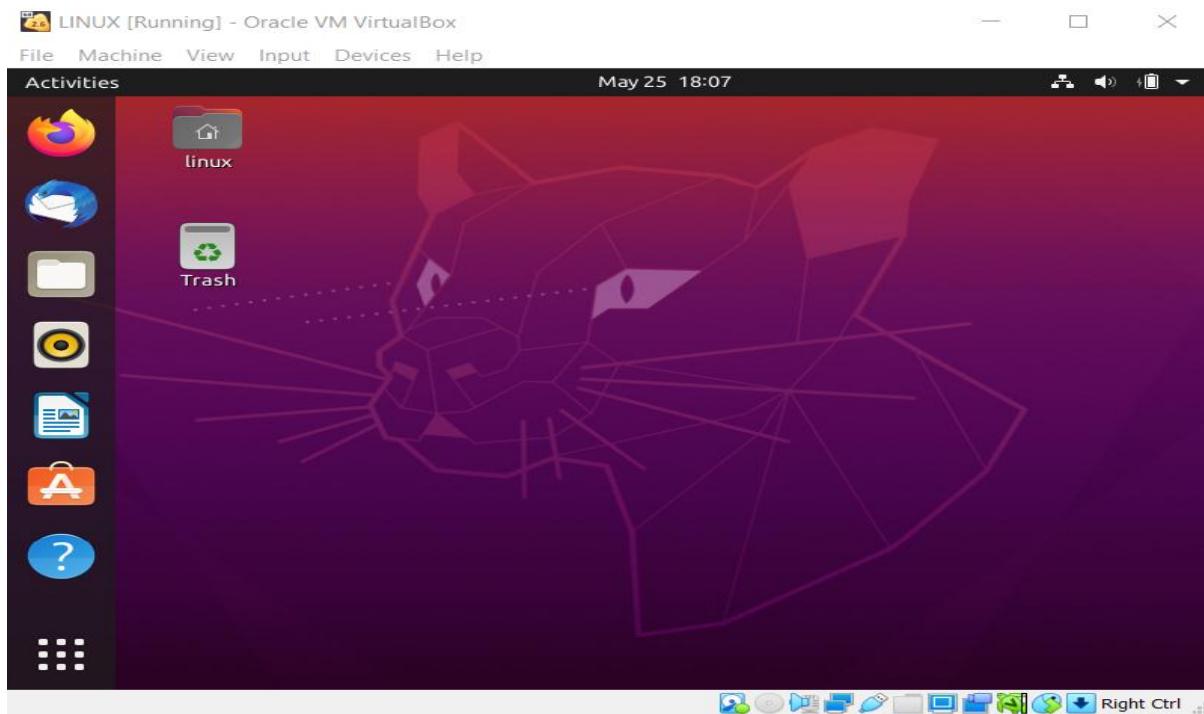
**Step -12)** Click on Restart Now



**Step-13)** After finishing the Restarting, you will see Ubuntu Desktop. Enter Password



### Step-14) ubuntu Desktop page



After installing OS, let's get started with some basic commands used in Linux.

## 2.3 Linux Commands

**Here is a list of basic Linux commands:**

### *1. pwd command*

Use the **pwd** command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash (**/**). An example of an absolute path is **/home/username**.

### *2. cd command*

To navigate through the Linux files and directories, use the **cd** command. It requires either the full path or the name of the directory, depending on the current working directory that you're in.

Let's say you're in **/home/username/Documents** and you want to go to **Photos**, a subdirectory of **Documents**. To do so, simply type the following command: **cd Photos**.

Another scenario is if you want to switch to a completely new directory, for example, **/home/username/Movies**. In this case, you have to type **cd** followed by the directory's absolute path: **cd /home/username/Movies**.

There are some shortcuts to help you navigate quickly:

**cd ..** (with two dots) to move one directory up

**cd** to go straight to the home folder

**cd-** (with a hyphen) to move to your previous directory

On a side note, Linux's shell is case sensitive. So, you have to type the name's directory exactly as it is.

### **3. ls command**

The **ls** command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.

If you want to see the content of other directories, type **ls** and then the directory's path. For example, enter **ls /home/username/Documents** to view the content of **Documents**.

There are variations you can use with the **ls** command:

**ls -R** will list all the files in the sub-directories as well

**ls -a** will show the hidden files

**ls -al** will list the files and directories with detailed information like the permissions, size, owner, etc.

### **4. cat command**

**cat** (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (stdout). To run this command, type **cat** followed by the file's name and its extension. For instance: **cat file.txt**.

Here are other ways to use the **cat** command:

**cat > filename** creates a new file

**cat filename1 filename2>filename3** joins two files (1 and 2) and stores the output of them in a new file (3)

to convert a file to upper- or lower-case use, **cat filename | tr a-z A-Z >output.txt**

### **5. cp command**

Use the **cp** command to copy files from the current directory to a different directory. For instance, the command **cp scenery.jpg /home/username/Pictures** would create a copy of **scenery.jpg** (from your current directory) into the **Pictures** directory.

### **6. mv command**

The primary use of the **mv** command is to move files, although it can also be used to rename files.

The arguments in mv are similar to the cp command. You need to type **mv**, the file's name, and the destination's directory. For example: **mv file.txt /home/username/Documents**.

To rename files, the Linux command is **mv oldname.ext newname.ext**

### **7. mkdir command**

Use **mkdir** command to make a new directory — if you type **mkdir Music** it will create a directory called **Music**.

There are extra **mkdir** commands as well:

To generate a new directory inside another directory, use this Linux basic command **mkdir Music/Newfile**

use the **p** (parents) option to create a directory in between two existing directories. For example, **mkdir -p Music/2020/Newfile** will create the new “2020” file.

### **8. rmdir command**

If you need to delete a directory, use the **rmdir** command. However, rmdir only allows you to delete empty directories.

### **9. rm command**

The **rm** command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to rmdir — use **rm -r**.

**Note:** Be very careful with this command and double-check which directory you are in. This will delete everything and there is no undo.

### **10. touch command**

The **touch** command allows you to create a blank new file through the Linux command line. As an example, enter **touch /home/username/Documents/Web.html** to create an HTML file entitled **Web** under the **Documents** directory.

### **11. locate command**

You can use this command to **locate** a file, just like the search command in Windows. What's more, using the **-i** argument along with this command will make it case-insensitive, so you can search for a file even if you don't remember its exact name.

To search for a file that contains two or more words, use an asterisk (\*). For example, **locate -i school\*note** command will search for any file that contains the word “school” and “note”, whether it is uppercase or lowercase.

### **12. find command**

Similar to the **locate** command, using **find** also searches for files and directories. The difference is, you use the **find** command to locate files within a given directory.

As an example, find **/home/ -name notes.txt** command will search for a file called **notes.txt** within the home directory and its subdirectories.

Other variations when using the **find** are:

To find files in the current directory use, **find . -name notes.txt**

To look for directories use, **/ -type d -name notes. txt**

### **13. grep command**

Another basic Linux command that is undoubtedly helpful for everyday use is **grep**.

It lets you search through all the text in a given file.

To illustrate, **grep blue notepad.txt** will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully.

### **14. sudo command**

Short for “**SuperUser Do**”, this command enables you to perform tasks that require administrative or root permissions. However, it is not advisable to use this command for daily use because it might be easy for an error to occur if you did something wrong.

### **15. df command**

Use **df** command to get a report on the system’s disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type **df -m**.

### **16. du command**

If you want to check how much space a file or a directory takes, the **du** (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the **-h** argument to the command line.

### **17. head command**

The **head** command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. For example, if you only want to show the first five lines, type **head -n 5 filename.ext**.

### **18. tail command**

This one has a similar function to the head command, but instead of showing the first lines, the **tail** command will display the last ten lines of a text file. For example, **tail -n filename.ext**.

### **19. diff command**

Short for difference, the **diff** command compares the contents of two files line by line. After analyzing the files, it will output the lines that do not match. Programmers often use this command when they need to make program alterations instead of rewriting the entire source code.

The simplest form of this command is **diff file1.ext file2.ext**

#### **20. tar command**

The **tar** command is the most used command to archive multiple files into a **tarball** — a common Linux file format that is similar to zip format, with compression being optional.

This command is quite complex with a long list of functions such as adding new files into an existing archive, listing the content of an archive, extracting the content from an archive, and many more.

#### **21. chmod command**

**chmod** is another Linux command, used to change the read, write, and execute permissions of files and directories. As this command is rather complicated

#### **22. chown command**

In Linux, all files are owned by a specific user. The **chown** command enables you to change or transfer the ownership of a file to the specified username. For instance, **chown linuxuser2 file.ext** will make **linuxuser2** as the owner of the **file.ext**.

#### **23. jobs command**

**jobs** command will display all current jobs along with their statuses. A job is basically a process that is started by the shell.

#### **24. kill command**

If you have an unresponsive program, you can terminate it manually by using the **kill** command. It will send a certain signal to the misbehaving app and instructs the app to terminate itself.

There is a total of **sixty-four signals** that you can use, but people usually only use two signals:

**SIGTERM (15)** — requests a program to stop running and gives it some time to save all of its progress. If you don't specify the signal when entering the **kill** command, this signal will be used.

**SIGKILL (9)** — forces programs to stop immediately. Unsaved progress will be lost. Besides knowing the signals, you also need to know the process identification number (PID) of the program you want to **kill**. If you don't know the PID, simply run the command **ps ux**.

After knowing what signal you want to use and the PID of the program, enter the following syntax:

**kill [signal option] PID.**

### **25. ping command**

Use the **ping** command to check your connectivity status to a server. For example, by simply entering **ping google.com**, the command will check whether you're able to connect to Google and also measure the response time.

### **26. wget command**

The Linux command line is super useful — you can even download files from the internet with the help of the **wget** command. To do so, simply type **wget** followed by the download link.

### **27. uname command**

The **uname** command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.

### **28. top command**

As a terminal equivalent to Task Manager in Windows, the **top** command will display a list of running processes and how much CPU each process uses. It's very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.

### **29. history command**

When you've been using Linux for a certain period of time, you'll quickly notice that you can run hundreds of commands every day. As such, running **history** command is particularly useful if you want to review the commands you've entered before.

### **30. man command**

Confused about the function of certain Linux commands? Don't worry, you can easily learn how to use them right from Linux's shell by using the **man** command. For instance, entering **man tail** will show the manual instruction of the tail command.

### **31. echo command**

This command is used to move some data into a file. For example, if you want to add the text, "Hello, my name is John" into a file called **name.txt**, you would type **echo Hello, my name is John >> name.txt**

### **32. zip, unzip command**

Use the **zip** command to compress your files into a zip archive, and use the **unzip** command to extract the zipped files from a zip archive.

### **33. hostname command**

If you want to know the name of your host/network simply type **hostname**. Adding a **-I** to the end will display the IP address of your network.

### 34. **useradd, userdel command**

Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time. **useradd** is used to create a new user, while **passwd** is adding a password to that user's account. To add a new person named John type, **useradd John** and then to add his password type, **passwd 123456789**.

To remove a user is very similar to adding a new user. To delete the users account type, **userdel UserName [13]**

Command	Description
ls	Lists all files and directories in the present working directory
ls - R	Lists files in sub-directories as well
ls - a	Lists hidden files as well
ls - al	Lists files and directories with detailed information like permissions, size, owner, etc.
cat > filename	Creates a new file
cat filename	Displays the file content
cat file1 file2 > file3	Joins two files (file1, file2) and stores the output in a new file (file3)
mv file "new file path"	Moves the files to the new location
mv filename new_file_name	Renames the file to a new filename
sudo	Allows regular users to run programs with the security privileges of the superuser or root
rm filename	Deletes a file
man	Gives help information on a command

history	Gives a list of all past basic Linux commands list typed in the current terminal session
clear	Clears the terminal
mkdir directoryname	Creates a new directory in the present working directory or a at the specified path
Rmdir	Deletes a directory
Mv	Renames a directory
pr -x	Divides the file into x columns
pr -h	Assigns a header to the file
pr -n	Denotes the file with Line Numbers
lp -nc lpr c	Prints "c" copies of the File
lp -d lpr -P	Specifies name of the printer
apt-get	Command used to install and update packages
mail -s 'subject' -c 'cc-address' -b 'bcc-address' 'to-address'	Command to send email
mail -s "Subject" to-address < Filename	Command to send email with attachment

## 2.4 Shell Scripting

**Shell Scripting** is an open-source computer program designed to be run by the Unix/Linux shell. Shell Scripting is a program to write a series of commands for the shell to execute. It can combine lengthy and repetitive sequences of commands into a single and simple script that can be stored and executed anytime which, reduces programming efforts.

## Commands

### CAT

Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files. So let us see some frequently used cat commands.

#### 1) To view a single file

**Command:**

\$cat filename

Output

It will show content of given filename

#### 2) To view multiple files

**Command:**

\$cat file1 file2

Output

This will show the content of file1 and file2.

#### 3) To view contents of a file preceding with line numbers.

**Command:**

\$cat -n filename

Output

It will show content with line number

example:-cat-n Language.txt

1)This is Langauge

2)A unique array

#### 4) Create a file

**Command:**

\$ cat >newfile

Output

Will create and a file named newfile

#### 5) Copy the contents of one file to another file.

**Command:**

\$cat [filename-whose-contents-is-to-be-copied] > [destination-filename]

Output

The content will be copied in destination file

#### 6) Cat command can suppress repeated empty lines in output

**Command:**

\$cat -s Language.txt

Output

Will suppress repeated empty lines in output

**7) Cat command can append the contents of one file to the end of another file.**

**Command:**

\$cat file1 >> file2

Output

Will append the contents of one file to the end of another file

**8) Cat command can display content in reverse order using tac command.**

**Command:**

\$tac filename

Output

Will display content in reverse order

**9) Cat command can highlight the end of line.**

**Command:**

\$cat -E "filename"

Output

Will highlight the end of line

**10) If you want to use the -v, -E and -T option together, then instead of writing -vET in the command, you can just use the -A command line option.**

Command

\$cat -A "filename"

**11) Cat command to open dashed files.**

**Command:**

\$cat -- "-dashfile"

Output

Will display the content of -dashfile

**12) Cat command if the file has a lot of content and can't fit in the terminal.**

**Command:**

\$cat "filename" | more

Output

Will show that much content, which could fit in terminal

**13) Cat command to merge the contents of multiple files.**

**Command:**

\$cat "filename1" "filename2" "filename3" > "merged\_filename"

Output

Will merge the contents of file in respective order and will insert that content in "merged\_filename".

#### 14) Cat command to display the content of all text files in the folder.

**Command:**

\$cat \*.txt

**Output**

Will show the content of all text files present in the folder.

#### 15) Cat command to write in an already existing file.

**Command :**

\$cat >> Languages.txt

The newly added text.

**Output**

Will append the text "The newly added text." to the end of the file.

## 2.5 Working on different text editors: nano, vi

### Nano Editor

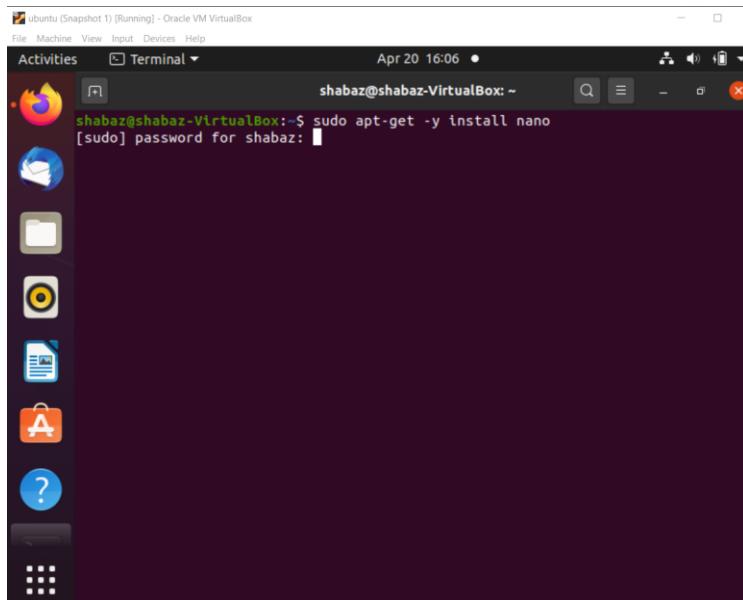
GNU nano is easy to use command line text editor for Linux operating systems. It includes all the basic functionality you'd expect from a regular text editor, like syntax highlighting, multiple buffers, search and replace with regular expression support, spellchecking, UTF-8 encoding, and more.

#### *Install NANO editor*

**Command:**

sudo apt-get -y install nano

**Output:**

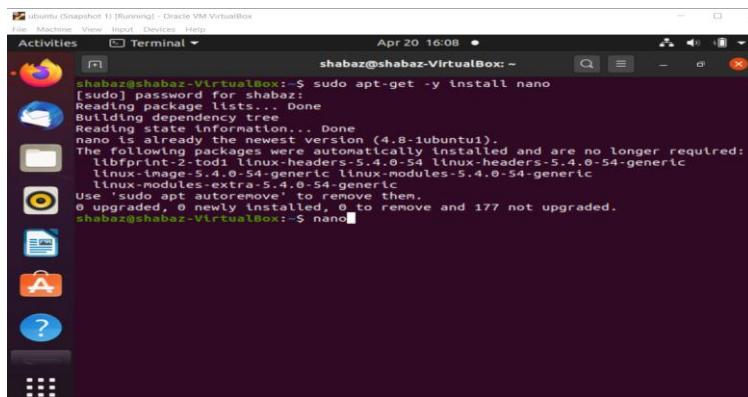


Enter the Password

**Command:**

sudo apt-get -y install nano

**Output:**



**Command:**

echo "what is your name?"

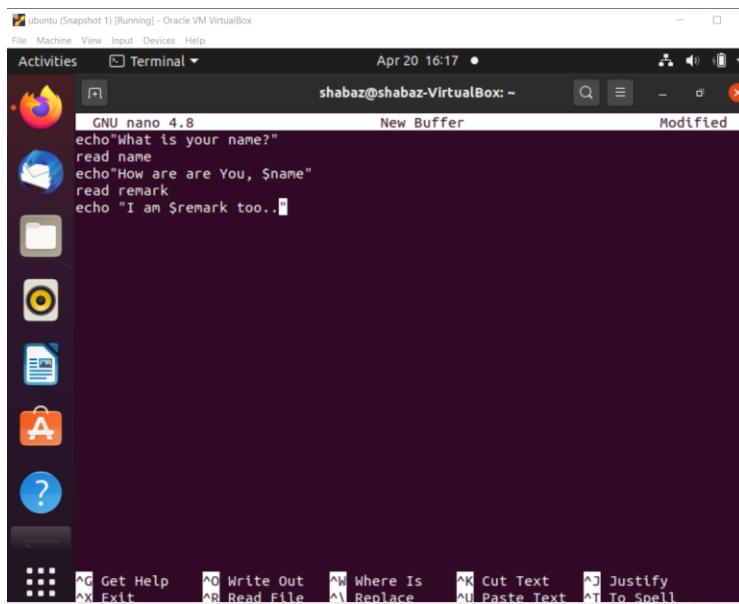
read name

echo "How do you do, \$name?"

read remark

echo "I am \$remark too!"

**Output:**



```
shabaz@shabaz-VirtualBox: ~
```

```
GNU nano 4.8          New Buffer          Modified
```

```
echo"What is your name?"
```

```
read name
```

```
echo"How are are You, $name"
```

```
read remark
```

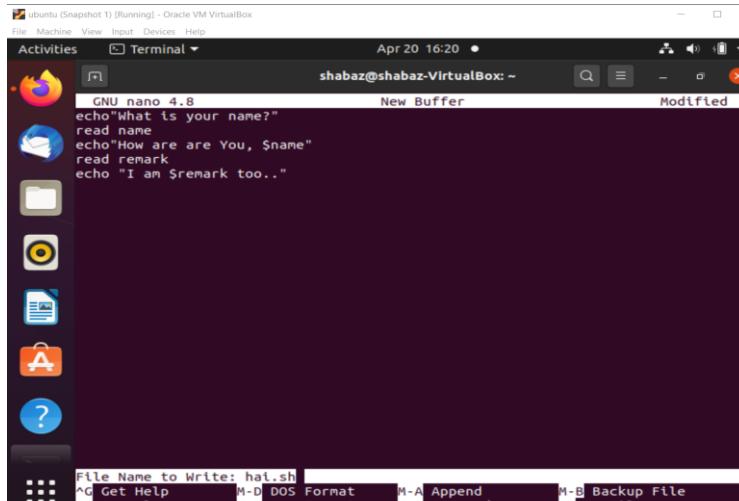
```
echo "I am $remark too.."
```

Click **CRTL+X**

Enter **Y**

Enter **FileName.sh**

Click **Enter**



```
shabaz@shabaz-VirtualBox: ~
```

```
GNU nano 4.8          New Buffer          Modified
```

```
echo"What is your name?"
```

```
read name
```

```
echo"How are are You, $name"
```

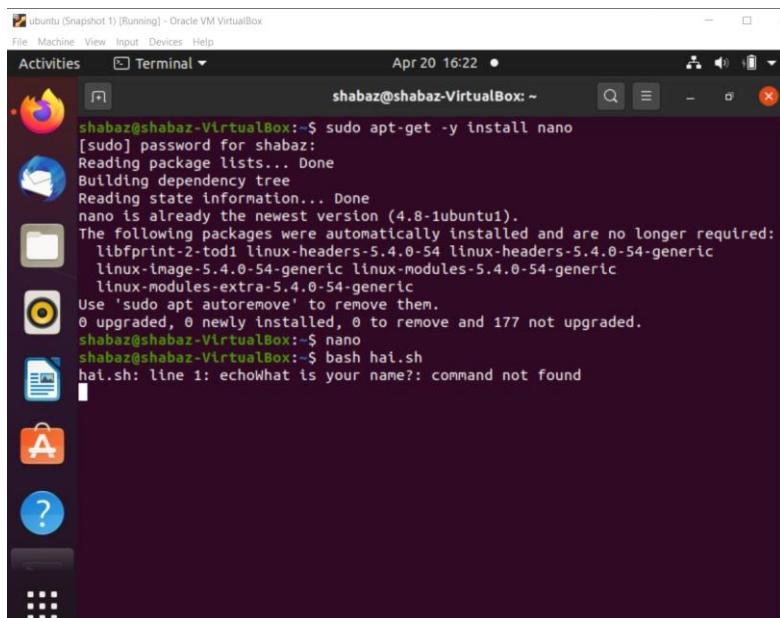
```
read remark
```

```
echo "I am $remark too.."
```

**Command:**

bash filename.sh

**Output:**



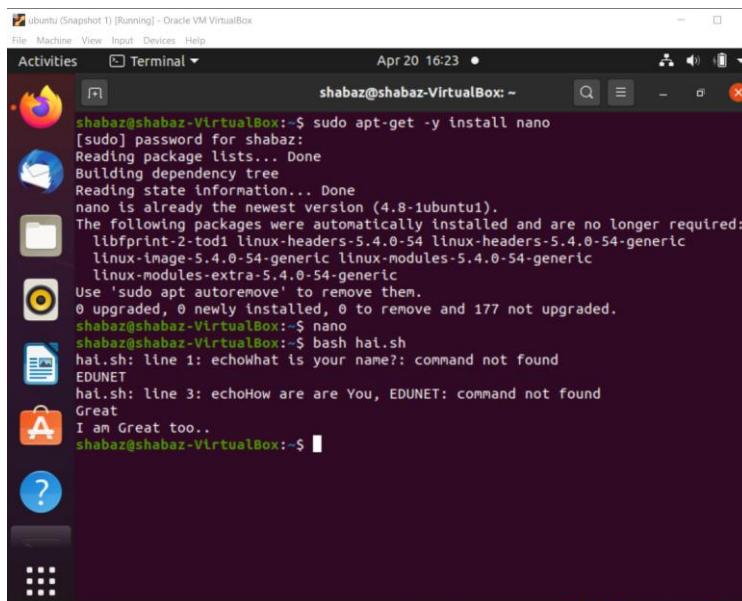
ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Apr 20 16:22

```
shabaz@shabaz-VirtualBox:~$ sudo apt-get -y install nano
[sudo] password for shabaz:
Reading package lists... Done
Building dependency tree
Reading state information... Done
nano is already the newest version (4.8-1ubuntu1).
The following packages were automatically installed and are no longer required:
libfprint-2-tod1 linux-headers-5.4.0-54 linux-headers-5.4.0-54-generic
linux-image-5.4.0-54-generic linux-modules-5.4.0-54-generic
linux-modules-extra-5.4.0-54-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 177 not upgraded.
shabaz@shabaz-VirtualBox:~$ nano
shabaz@shabaz-VirtualBox:~$ bash hai.sh
hai.sh: line 1: echoWhat is your name?: command not found
```

## Result



ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Apr 20 16:23

```
shabaz@shabaz-VirtualBox:~$ sudo apt-get -y install nano
[sudo] password for shabaz:
Reading package lists... Done
Building dependency tree
Reading state information... Done
nano is already the newest version (4.8-1ubuntu1).
The following packages were automatically installed and are no longer required:
libfprint-2-tod1 linux-headers-5.4.0-54 linux-headers-5.4.0-54-generic
linux-image-5.4.0-54-generic linux-modules-5.4.0-54-generic
linux-modules-extra-5.4.0-54-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 177 not upgraded.
shabaz@shabaz-VirtualBox:~$ nano
shabaz@shabaz-VirtualBox:~$ bash hai.sh
hai.sh: line 1: echoWhat is your name?: command not found
EDUNET
hat.sh: line 3: echoHow are are You, EDUNET: command not found
Great
I am Great too..
shabaz@shabaz-VirtualBox:~$
```

## VI Editor

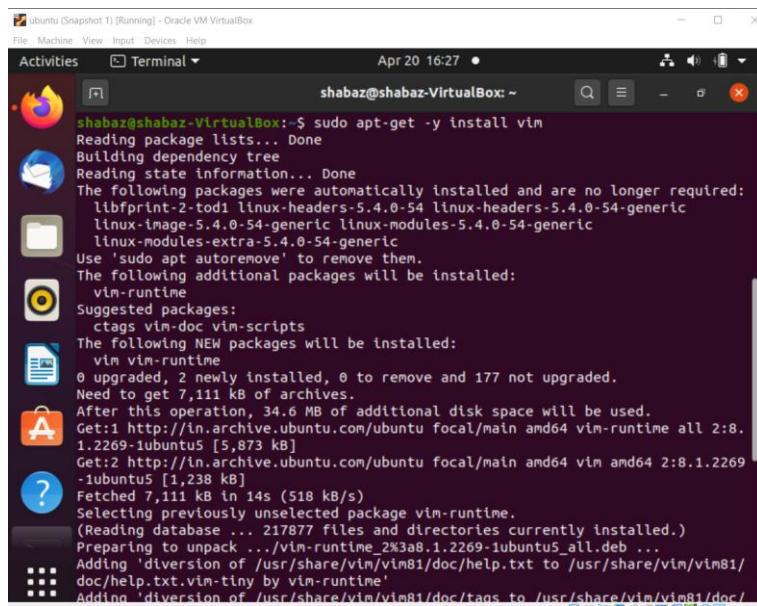
The VI editor is the most popular and classic text editor in the Linux family. It is available in almost all Linux distributions. The latest version is vim [ vi improved].

### Install VI editor

#### Command:

```
sudo apt-get -y install vim
```

#### Output:

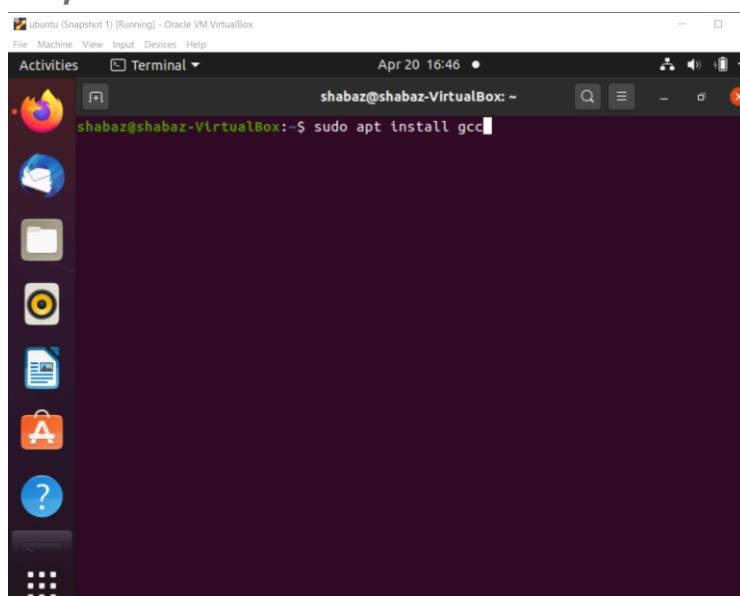


```
ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 20 16:27 •
shabaz@shabaz-VirtualBox:~$ sudo apt-get -y install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
libbfprint-2-tod1 linux-headers-5.4.0-54 linux-headers-5.4.0-54-generic
linux-image-5.4.0-54-generic linux-modules-5.4.0-54-generic
linux-modules-extra-5.4.0-54-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
vim-runtime
Suggested packages:
ctags vim-doc vim-scripts
The following NEW packages will be installed:
vim vim-runtime
0 upgraded, 2 newly installed, 0 to remove and 177 not upgraded.
Need to get 7,111 kB of archives.
After this operation, 34.6 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 vim-runtime all 2:8.1.2269-1ubuntu5 [5,873 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 vim amd64 2:8.1.2269-1ubuntu5 [1,238 kB]
Fetched 7,111 kB in 14s (518 kB/s)
Selecting previously unselected package vim-runtime.
(Reading database ... 217877 files and directories currently installed.)
Preparing to unpack .../vim-runtime_2%3a8.1.2269-1ubuntu5_all.deb ...
Adding 'diversion of /usr/share/vim/vim81/doc/help.txt to /usr/share/vim/vim81/doc/help.txt.vim-tiny by vim-runtime'
Adding 'diversion of /usr/share/vim/vim81/doc/tags to /usr/share/vim/vim81/doc/
```

**Command:**

```
sudo apt install gcc
```

**Output:**



```
ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 20 16:46 •
shabaz@shabaz-VirtualBox:~$ sudo apt install gcc
```

Enter the Password

**Command:**

```
vim welcome.c
```

Insert "I" to start the coding

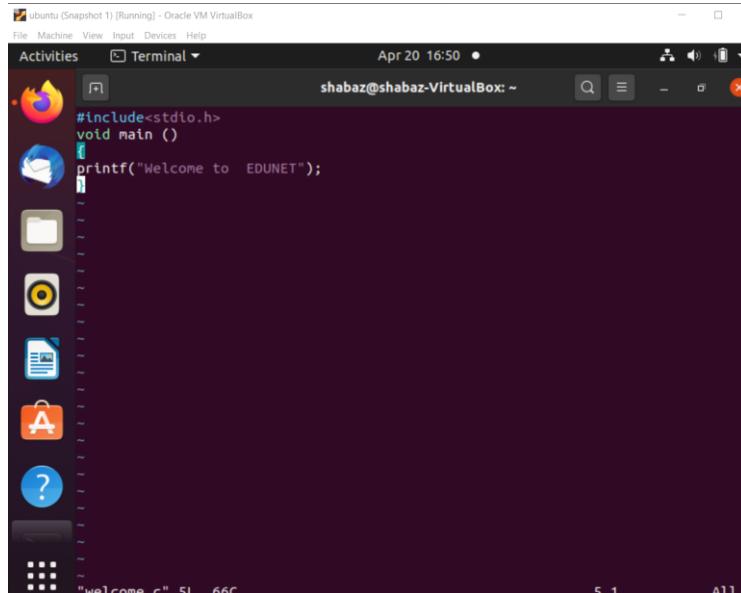
**Code**

```
#include<stdio.h>
void main()
{
printf("Welcome EDUNET");
```

}

**EXIT Click on “ESC” & Enter “:wq”**

**Output:**

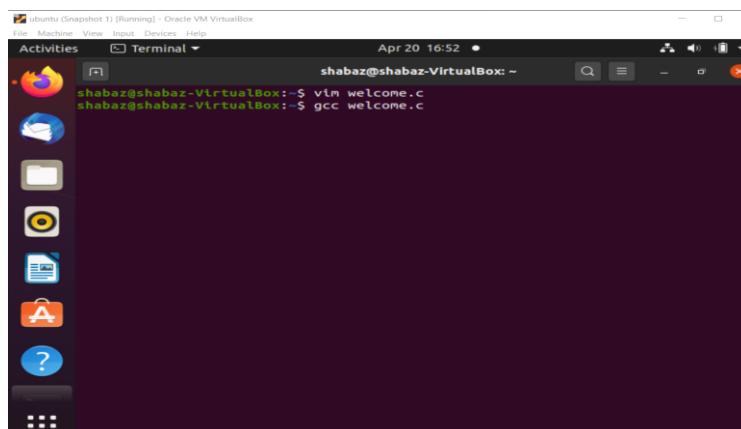


```
#include<stdio.h>
void main ()
{
    printf("Welcome to EDUNET");
}
```

**Command:**

gcc filename.c

**Output:**

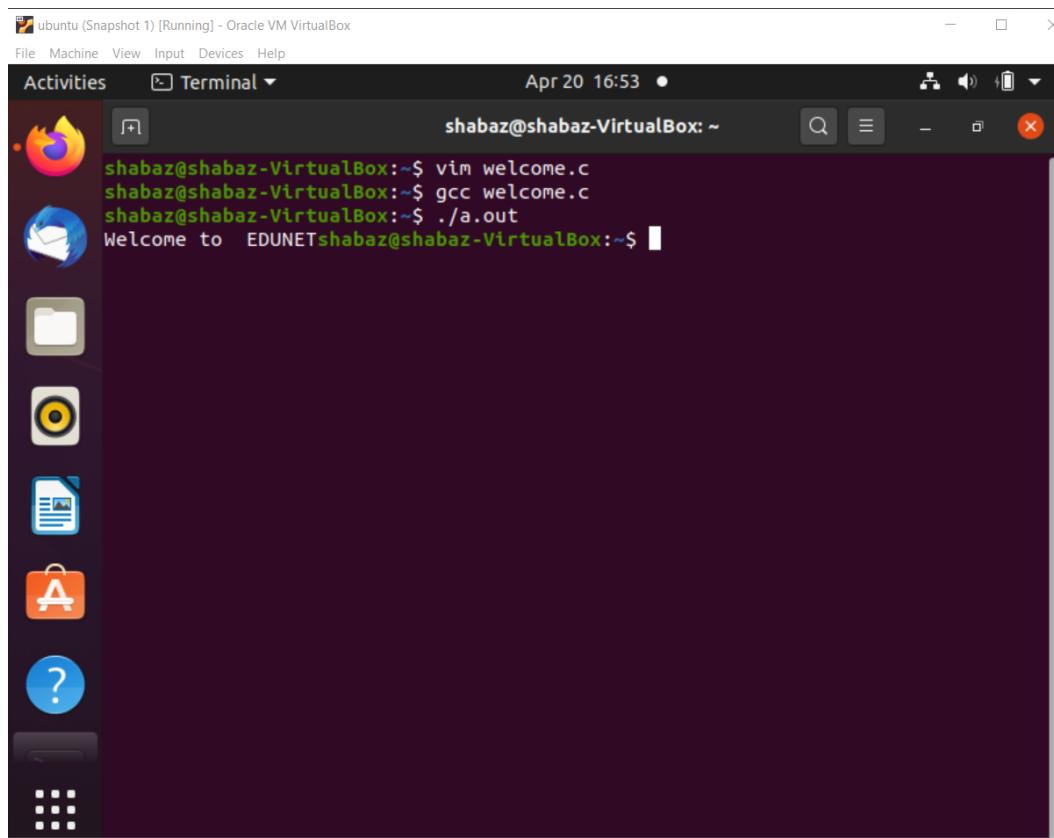


```
shabaz@shabaz-VirtualBox:~$ vim welcome.c
shabaz@shabaz-VirtualBox:~$ gcc welcome.c
```

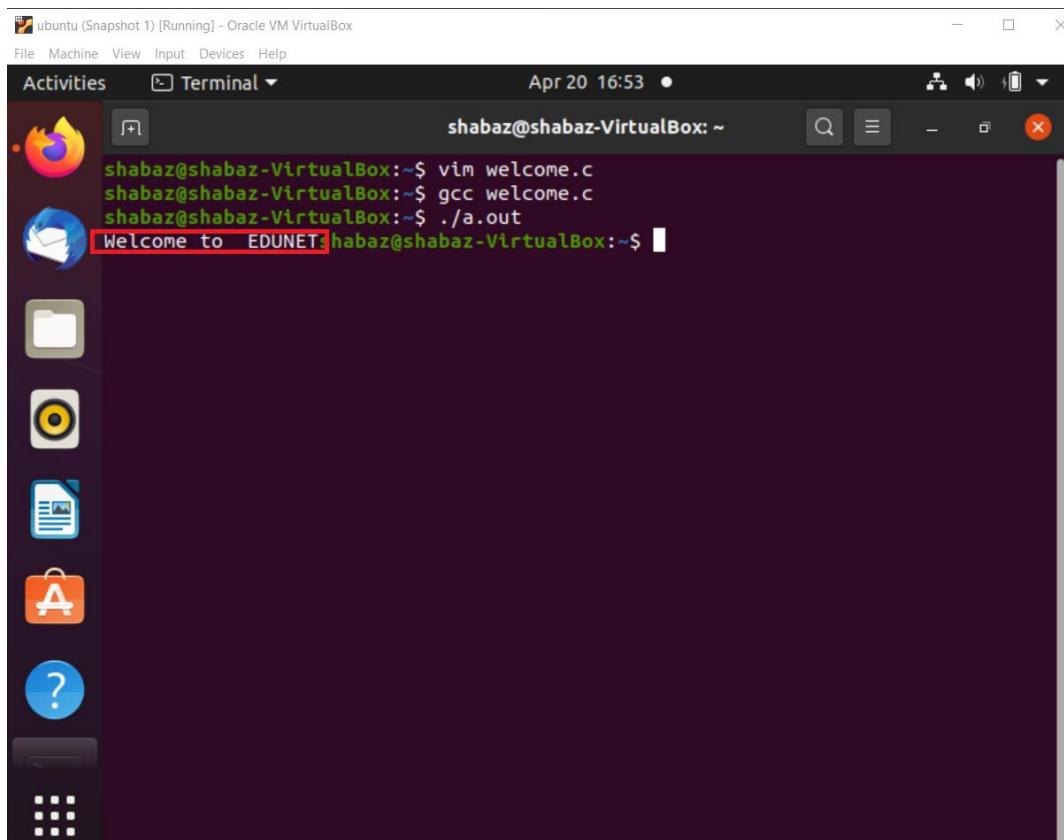
**Command:**

./a.out

**Output:**



*Result:*



## 2.6 Managing Linux Files

### Commands

#### CAT

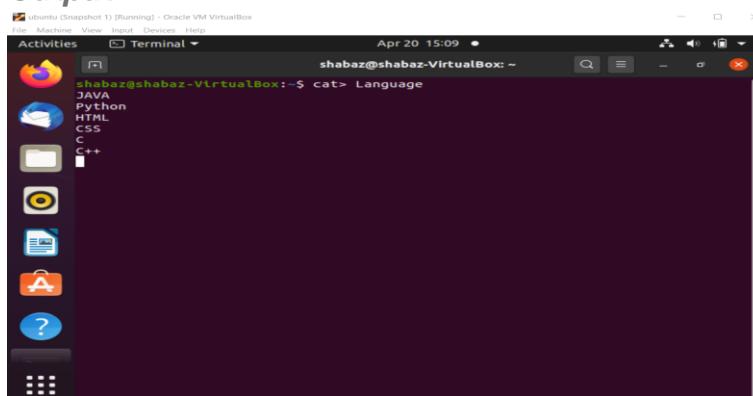
It is generally used to concatenate the files. It gives the output on the standard output.

#### Creating a file

##### *Command:*

cat > Filename

##### *Output:*



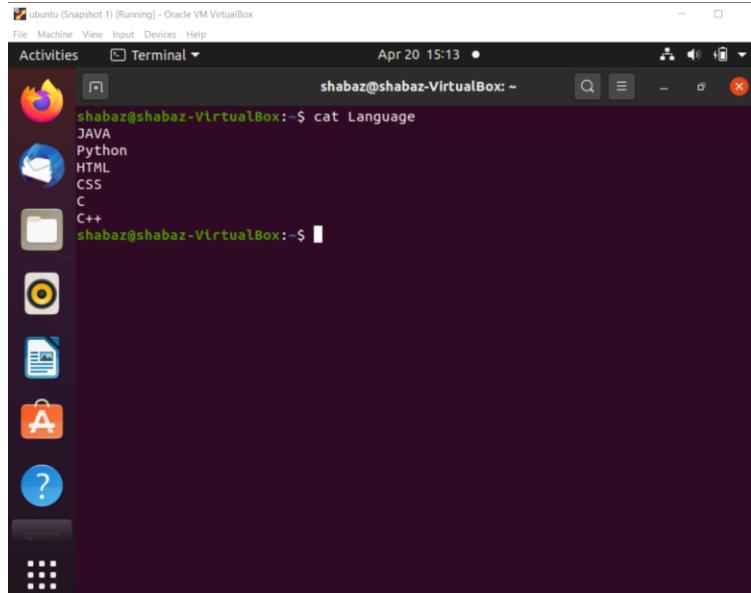
Entering the values to exit Press **CTRL+D**

## Calling a file

**Command:**

cat Filename

**Output**



A screenshot of a terminal window titled "ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox". The window shows the command "shabaz@shabaz-VirtualBox:~\$ cat Language" followed by the text "JAVA Python HTML CSS C C++". The terminal has a dark background with light-colored text. The window title bar includes the date and time: "Apr 20 15:13".

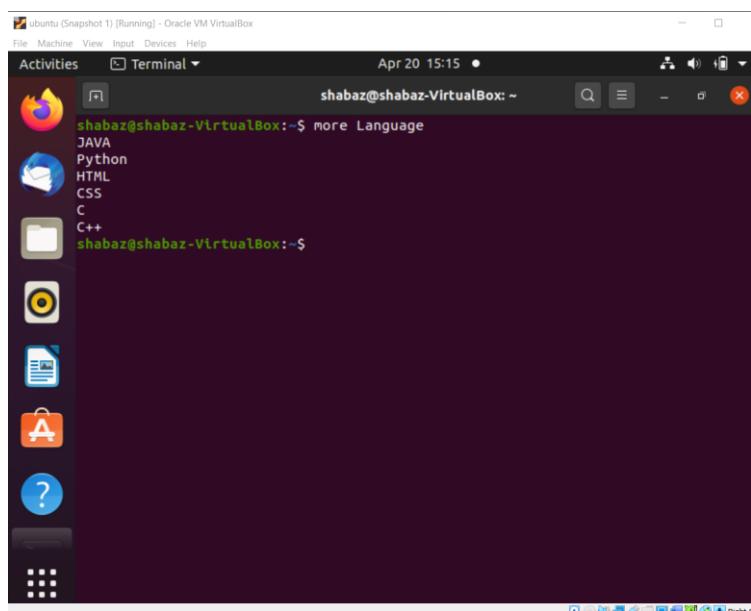
## More

It is a filter for paging through text one screenful at a time.

**Command:**

more Filename

**Output**



A screenshot of a terminal window titled "ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox". The window shows the command "shabaz@shabaz-VirtualBox:~\$ more Language" followed by the text "JAVA Python HTML CSS C C++". The terminal has a dark background with light-colored text. The window title bar includes the date and time: "Apr 20 15:15".

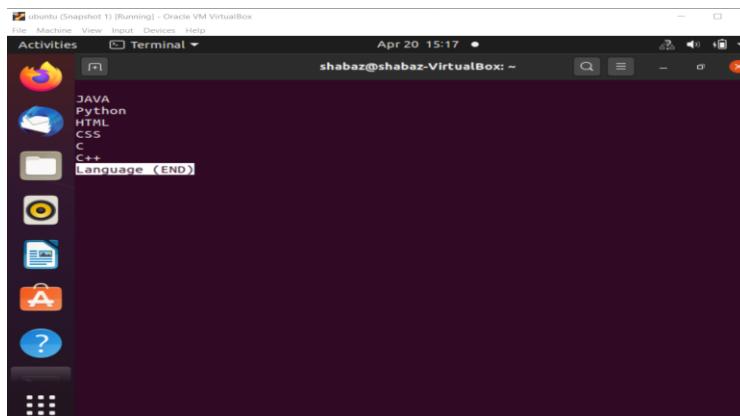
## Less

It is used to viewing the files instead of opening the file. Similar to *more* command but it allows backward as well as forward movement.

### *Command:*

less Filename

### *Output:*



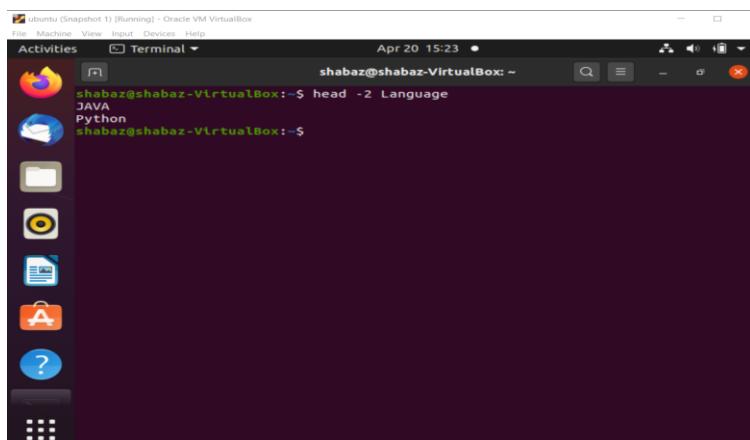
## Head

Used to print the first N lines of a file. It accepts N as input and the default value of N is 10.

### *Command:*

head -2 Filename

### *Output:*

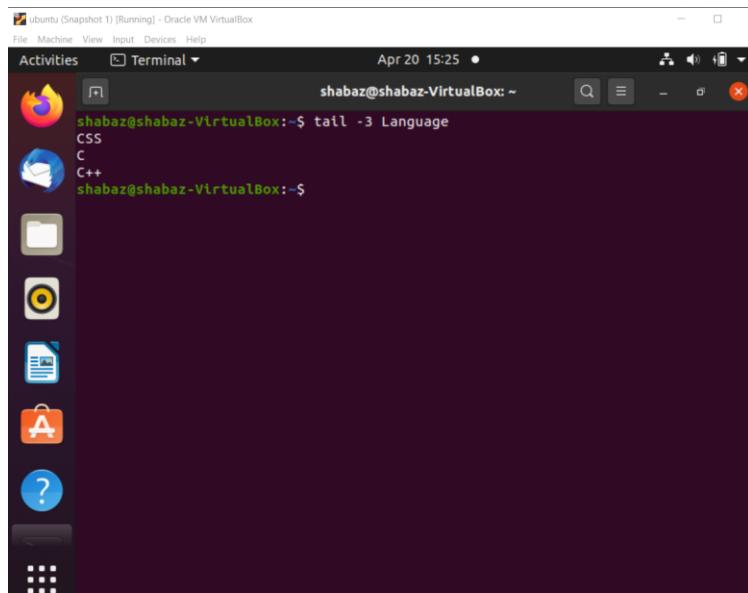


## Tail

Used to print the last N-1 lines of a file. It accepts N as input and the default value of N is 10.

### *Command:*

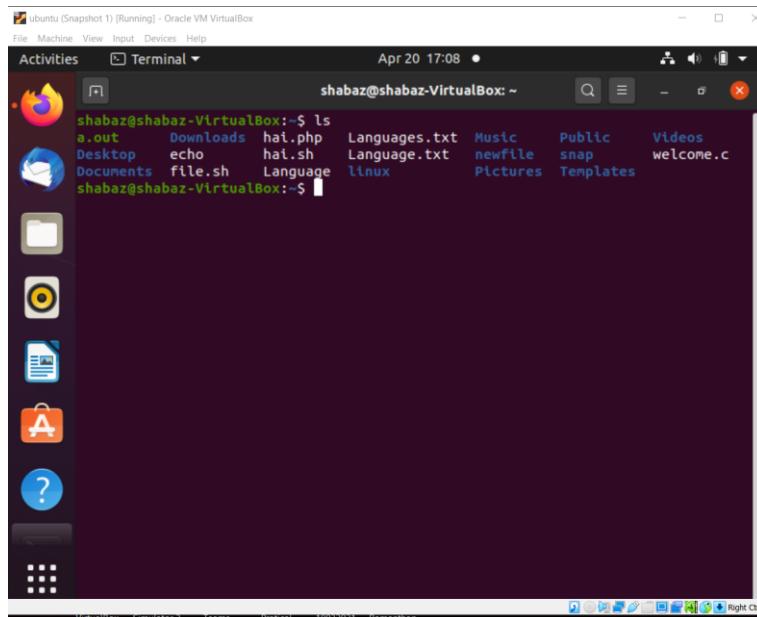
tail -3 Filename

**Output:**

```
ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 20 15:25 •
shabaz@shabaz-VirtualBox:~$ tail -3 Language
CSS
C
C++
shabaz@shabaz-VirtualBox:~$
```

**Files Listing**

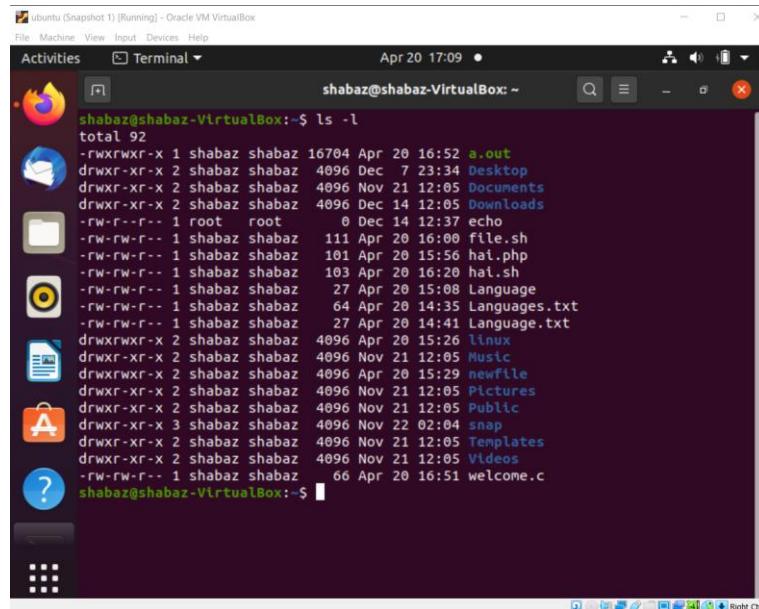
To perform Files listings or to list files and directories ls command is used

**Command:****ls****Output:**

```
ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 20 17:08 •
shabaz@shabaz-VirtualBox:~$ ls
a.out    Downloads  hal.php  Languages.txt  Music      Public    Videos
Desktop  echo      hal.sh   Language.txt  newfile   snap     welcome.c
Documents file.sh  Language  linux       Pictures  Templates
shabaz@shabaz-VirtualBox:~$
```

**Command:****ls -l**

## Output:



```
ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 20 17:09 •
shabaz@shabaz-VirtualBox:~$ ls -l
total 92
-rwxrwxr-x 1 shabaz shabaz 16704 Apr 20 16:52 a.out
drwxr-xr-x 2 shabaz shabaz 4096 Dec 7 23:34 Desktop
drwxr-xr-x 2 shabaz shabaz 4096 Nov 21 12:05 Documents
drwxr-xr-x 2 shabaz shabaz 4096 Dec 14 12:05 Downloads
-rw-r--r-- 1 root root 0 Dec 14 12:37 echo
-rw-rw-r-- 1 shabaz shabaz 111 Apr 20 16:00 file.sh
-rw-rw-r-- 1 shabaz shabaz 101 Apr 20 15:56 hal.php
-rw-rw-r-- 1 shabaz shabaz 103 Apr 20 16:20 hal.sh
-rw-rw-r-- 1 shabaz shabaz 27 Apr 20 15:08 Language
-rw-rw-r-- 1 shabaz shabaz 64 Apr 20 14:35 Languages.txt
-rw-rw-r-- 1 shabaz shabaz 27 Apr 20 14:41 Language.txt
drwxrwxr-x 2 shabaz shabaz 4096 Apr 20 15:26 linux
drwxr-xr-x 2 shabaz shabaz 4096 Nov 21 12:05 Music
drwxrwxr-x 2 shabaz shabaz 4096 Apr 20 15:29 newfile
drwxr-xr-x 2 shabaz shabaz 4096 Nov 21 12:05 Pictures
drwxr-xr-x 2 shabaz shabaz 4096 Nov 21 12:05 Public
drwxr-xr-x 3 shabaz shabaz 4096 Nov 22 02:04 snap
drwxr-xr-x 2 shabaz shabaz 4096 Nov 21 12:05 Templates
drwxr-xr-x 2 shabaz shabaz 4096 Nov 21 12:05 Videos
-rw-rw-r-- 1 shabaz shabaz 66 Apr 20 16:51 welcome.c
shabaz@shabaz-VirtualBox:~$
```

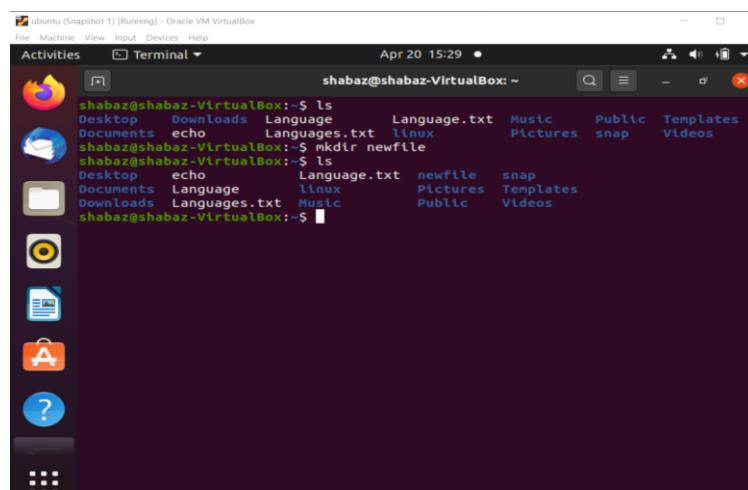
## MKDIR

Used to create a directory if not already exist. It accepts the directory name as an input parameter.

### Command:

```
mkdir newfile
```

## Output:



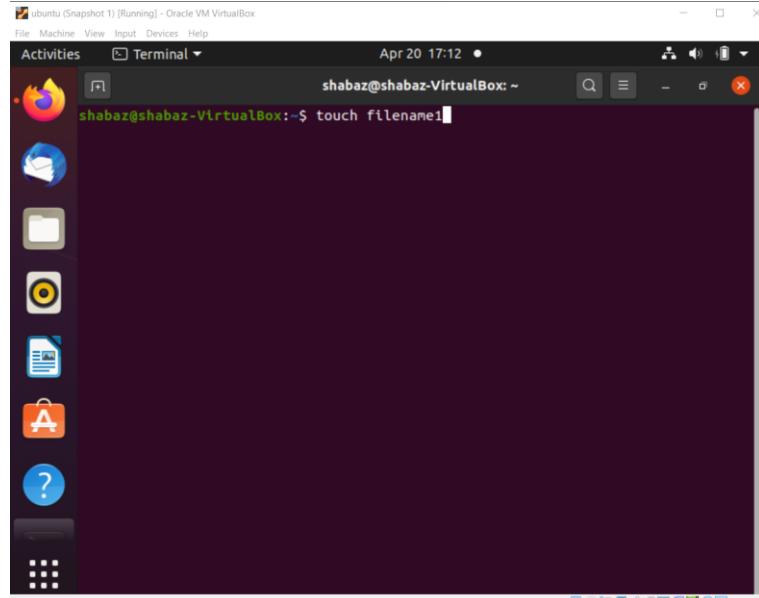
```
ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 20 15:29 •
shabaz@shabaz-VirtualBox:~$ ls
Desktop Documents Downloads Language Language.txt Music Public Templates
Documents echo Languages.txt linux Pictures snap Videos
shabaz@shabaz-VirtualBox:~$ mkdir newfile
shabaz@shabaz-VirtualBox:~$ ls
Desktop Documents echo Languages.txt newfile linux Pictures Templates
Downloads Languages.txt Music Public snap Videos
shabaz@shabaz-VirtualBox:~$
```

## Touch

Touch command can be used to create a new file. It will create and open a new blank file if the file with a filename does not exist. And in case the file already exists then the file will not be affected.

### Command:

```
touch newfile1
```

**Output:**

A screenshot of an Ubuntu desktop environment. A terminal window is open in the foreground, showing the command `shabaz@shabaz-VirtualBox:~\$ touch filename1` and its output. The desktop background is dark, and the Unity interface is visible.

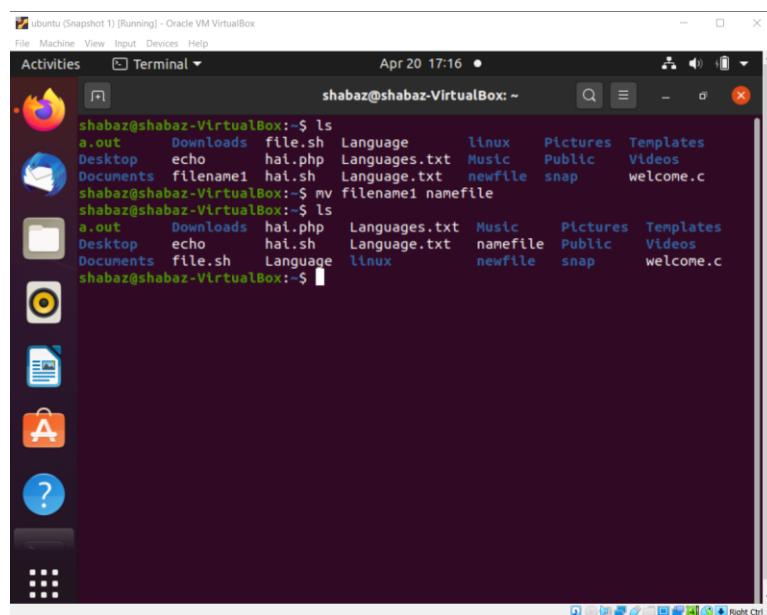
```
ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 20 17:12
shabaz@shabaz-VirtualBox:~$ touch filename1
```

## Renaming a File

mv command could be used to rename a file. It will rename the filename to new\_filename or in other words, it will remove the filename file and would be creating a new file with the new\_filename with the same content and name as that of the filename file.

**Command:**

```
mv filename1 namefile
```

**Output:**

A screenshot of an Ubuntu desktop environment. A terminal window is open in the foreground, showing the command `shabaz@shabaz-VirtualBox:~\$ mv filename1 namefile` and its output. The desktop background is dark, and the Unity interface is visible.

```
ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 20 17:16
shabaz@shabaz-VirtualBox:~$ ls
a.out  Downloads  file.sh  Language  linux  Pictures  Templates
Desktop  echo  hai.php  Languages.txt  Music  Public  Videos
Documents  filename1  hai.sh  Language.txt  newfile  snap  welcome.c
shabaz@shabaz-VirtualBox:~$ mv filename1 namefile
shabaz@shabaz-VirtualBox:~$ ls
a.out  Downloads  hai.php  Languages.txt  Music  Pictures  Templates
Desktop  echo  hai.sh  Language.txt  namefile  Public  Videos
Documents  file.sh  Language  linux  newfile  snap  welcome.c
shabaz@shabaz-VirtualBox:~$
```

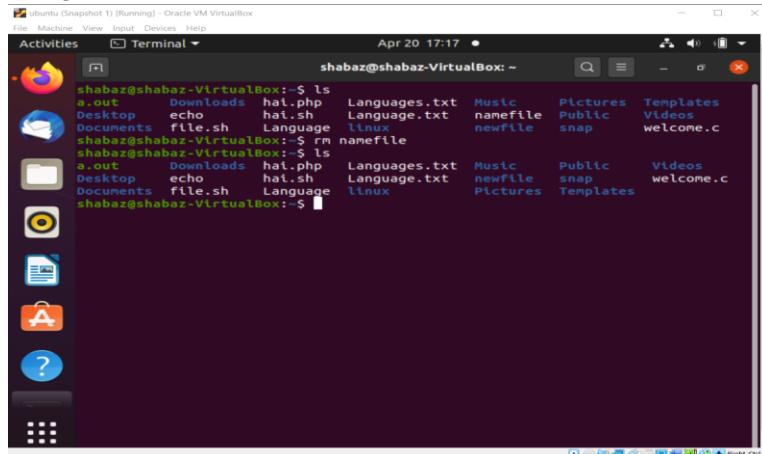
## Deleting a File

rm command could be used to delete a file. It will remove the filename file from the directory.

### Command:

rm filename

### Output:



```
ubuntu (Snapshot 1) (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 20 17:17
shabaz@shabaz-VirtualBox:~$ ls
a.out Downloads hal.php Languages.txt Music Pictures Templates
Desktop echo hal.sh Language.txt namefile Public Videos
Documents file.sh Language linux newfile snap
shabaz@shabaz-VirtualBox:~$ rm namefile
shabaz@shabaz-VirtualBox:~$ ls
a.out Downloads hal.php Languages.txt Music Pictures Templates
Desktop echo hal.sh Language.txt newfile Public Videos
Documents file.sh Language linux snap
shabaz@shabaz-VirtualBox:~$
```

## Managing User Permissions

There are three types of accounts on a Unix system –

### Root account

This is also called **superuser** and would have complete and unfettered control of the system. A superuser can run any commands without any restriction. This user should be assumed as a system administrator.

### System accounts

System accounts are those needed for the operation of system-specific components for example mail accounts and the **sshd** accounts. These accounts are usually needed for some specific function on your system, and any modifications to them could adversely affect the system.

### User accounts

User accounts provide interactive access to the system for users and groups of users. General users are typically assigned to these accounts and usually have limited access to critical system files and directories.

Unix supports a concept of *Group Account* which logically groups a number of accounts. Every account would be a part of another group account. A Unix group plays important role in handling file permissions and process management.

## Managing Users and Groups

There are four main user administration files –

- **/etc/passwd** – Keeps the user account and password information. This file holds the majority of information about accounts on the Unix system.

- **/etc/shadow** – Holds the encrypted password of the corresponding account. Not all the systems support this file.
- **/etc/group** – This file contains the group information for each account.
- **/etc/gshadow** – This file contains secure group account information.

Check all the above files using the **cat** command.

The following table lists out commands that are available on majority of Unix systems to create and manage accounts and groups –

Sr.No.	Command & Description
1	<b>useradd</b> Adds accounts to the system
2	<b>usermod</b> Modifies account attributes
3	<b>userdel</b> Deletes accounts from the system
4	<b>groupadd</b> Adds groups to the system
5	<b>groupmod</b> Modifies group attributes
6	<b>groupdel</b> Removes groups from the system

You can use Man page Help to check complete syntax for each command mentioned here.

### Create a Group

We will now understand how to create a group. For this, we need to create groups before creating any account otherwise, we can make use of the existing groups in our system. We have all the groups listed in **/etc/groups** file.

All the default groups are system account specific groups and it is not recommended to use them for ordinary accounts. So, following is the syntax to create a new group account –

### Command

`groupadd [-g gid [-o]] [-r] [-f] groupname`

The following table lists out the parameters –

Sr.No.	Option & Description
1	<b>-g GID</b> The numerical value of the group's ID
2	<b>-o</b> This option permits to add group with non-unique GID
3	<b>-r</b> This flag instructs <b>groupadd</b> to add a system account
4	<b>-f</b> This option causes to just exit with success status, if the specified group already exists. With -g, if the specified GID already exists, other (unique) GID is chosen
5	<b>groupname</b> Actual group name to be created

If you do not specify any parameter, then the system makes use of the default values.

Following example creates a *developers* group with default values, which is very much acceptable for most of the administrators.

**Command:**

```
$ groupadd developers
```

**Modify a Group**

To modify a group, use the **groupmod** syntax –

**Command:**

```
$ groupmod -n new_modified_group_name old_group_name
```

To change the *developers\_2* group name to *developer*, type –

**Command:**

```
$ groupmod -n developer developer_2
```

Here is how you will change the financial GID to 545 –

```
$ groupmod -g 545 developer
```

## Delete a Group

We will now understand how to delete a group. To delete an existing group, all you need is the **groupdel command** and the **group name**. To delete the financial group, the command is –

### *Command:*

```
$ groupdel developer
```

This removes only the group, not the files associated with that group. The files are still accessible by their owners.

## Create an Account

Let us see how to create a new account on your Unix system. Following is the syntax to create a user's account –

### *Command:*

```
useradd -d homedir -g groupname -m -s shell -u userid accountname
```

The following table lists out the parameters –

Sr.No.	Option & Description
1	<b>-d homedir</b> Specifies home directory for the account
2	<b>-g groupname</b> Specifies a group account for this account
3	<b>-m</b> Creates the home directory if it doesn't exist
4	<b>-s shell</b> Specifies the default shell for this account
5	<b>-u userid</b> You can specify a user id for this account
6	<b>accountname</b> Actual account name to be created

If you do not specify any parameter, then the system makes use of the default values. The **useradd** command modifies the **/etc/passwd**, **/etc/shadow**, and **/etc/group** files and creates a home directory.

Following is the example that creates an account **mcmohd**, setting its home directory to **/home/mcmohd** and the group as **developers**. This user would have Korn Shell assigned to it.

**Command:**

```
$ useradd -d /home/mcmohd -g developers -s /bin/ksh mcmohd
```

Before issuing the above command, make sure you already have the *developers* group created using the **groupadd** command.

Once an account is created you can set its password using the **passwd** command as follows –

**Command:**

```
$ passwd mcmohd20
```

Changing password for user mcmohd20.

New UNIX password:

Retype new UNIX password:

passwd: all authentication tokens updated successfully.

When you type **passwd account name**, it gives you an option to change the password, provided you are a superuser. Otherwise, you can change just your password using the same command but without specifying your account name.

## Modify an Account

The **usermod** command enables you to make changes to an existing account from the command line. It uses the same arguments as the **useradd** command, plus the **-l** argument, which allows you to change the account name.

For example, to change the account name **mcmohd** to **mcmohd20** and to change home directory accordingly, you will need to issue the following command –

**Command:**

```
$ usermod -d /home/mcmohd20 -m -l mcmohd mcmohd20
```

## Delete an Account

The **userdel** command can be used to delete an existing user. This is a very dangerous command if not used with caution.

There is only one argument or option available for the command **.r**, for removing the account's home directory and mail file.

For example, to remove account **mcmohd20**, issue the following command –

**Command:**

```
$ userdel -r mcmohd20
```

If you want to keep the home directory for backup purposes, omit the **-r** option. You can remove the home directory as needed at a later time.

So, after getting the knowledge about an open-source Linux operating system, its commands and shell scripting we will move forward to learn about some open-source software and programming language that are used by data scientists and machine learning engineers.

## SSH Command

ssh stands for “Secure Shell”. It is a protocol used to securely connect to a remote server/system.

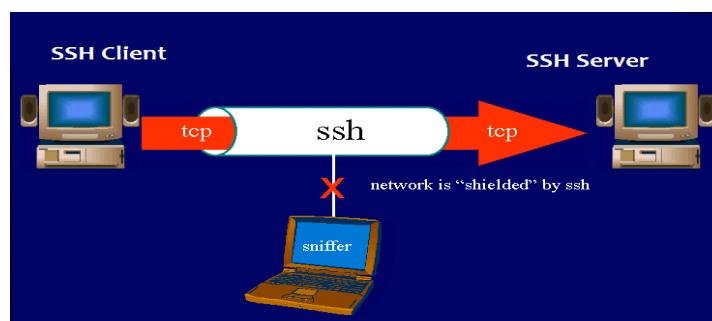
ssh is secure in the sense that it transfers the data in encrypted form between the host and the client.

It transfers inputs from the client to the host and relays back the output. ssh runs at TCP/IP port 22

Syntax -

```
ssh user_name@host(IP/Domain_name)
```

Example: ssh root@192.168.1.1



## SCP Command

The scp command allows you to copy files over ssh connections.

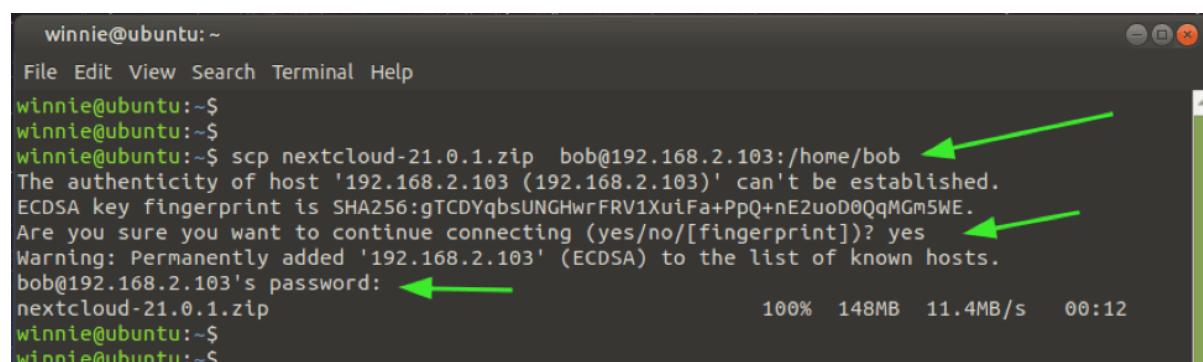
This is pretty useful if you want to transport files between computers

Syntax

```
scp examplefile yourusername@yourserver:/home/yourusername/
```

Example

```
scp file1.pdf root@192.168.1.1:/root/Desktop
```



```
winnie@ubuntu:~$ File Edit View Search Terminal Help
winnie@ubuntu:~$ winnie@ubuntu:~$ winnie@ubuntu:~$ scp nextcloud-21.0.1.zip bob@192.168.2.103:/home/bob
The authenticity of host '192.168.2.103 (192.168.2.103)' can't be established.
ECDSA key fingerprint is SHA256:gTCDYqbsUNGHwrFRV1XuiFa+PpQ+nE2uoD0QqMGm5WE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.2.103' (ECDSA) to the list of known hosts.
bob@192.168.2.103's password: 100% 148MB 11.4MB/s 00:12
nextcloud-21.0.1.zip
winnie@ubuntu:~$ winnie@ubuntu:~$
```

## 2.7 What is Data Analysis?

Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively.

Systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data. [ 3.15 ]

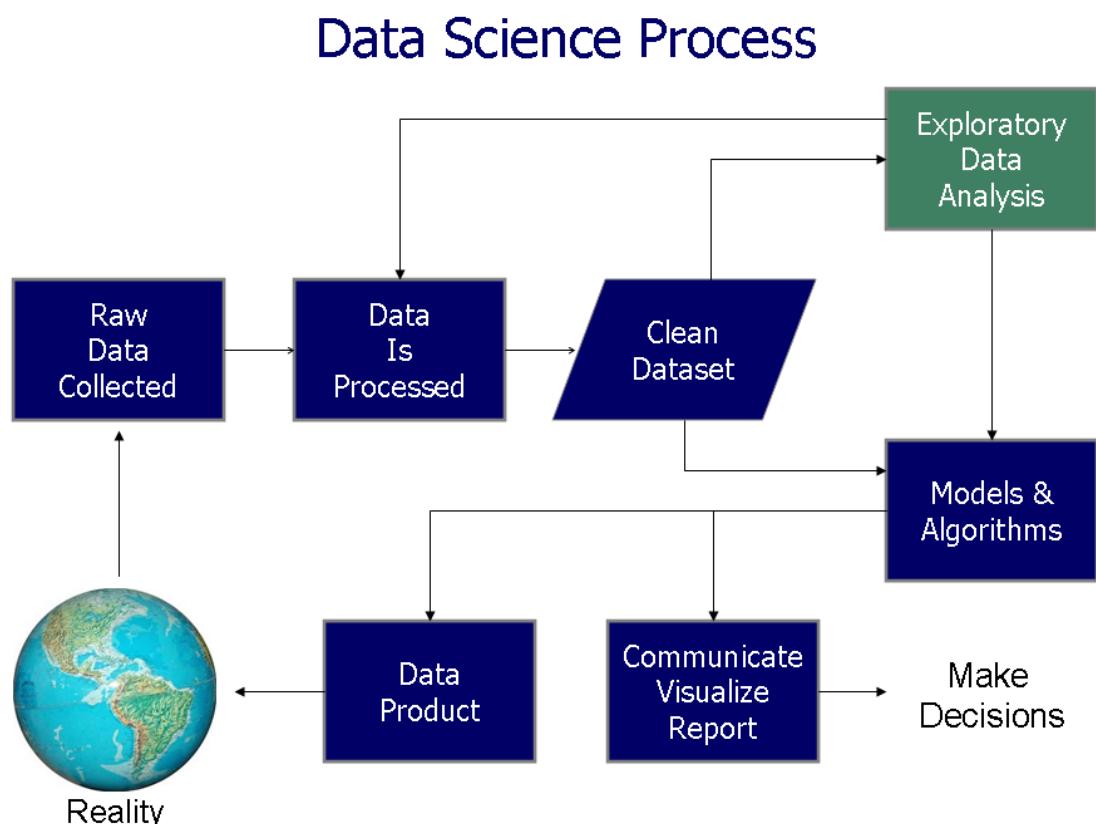
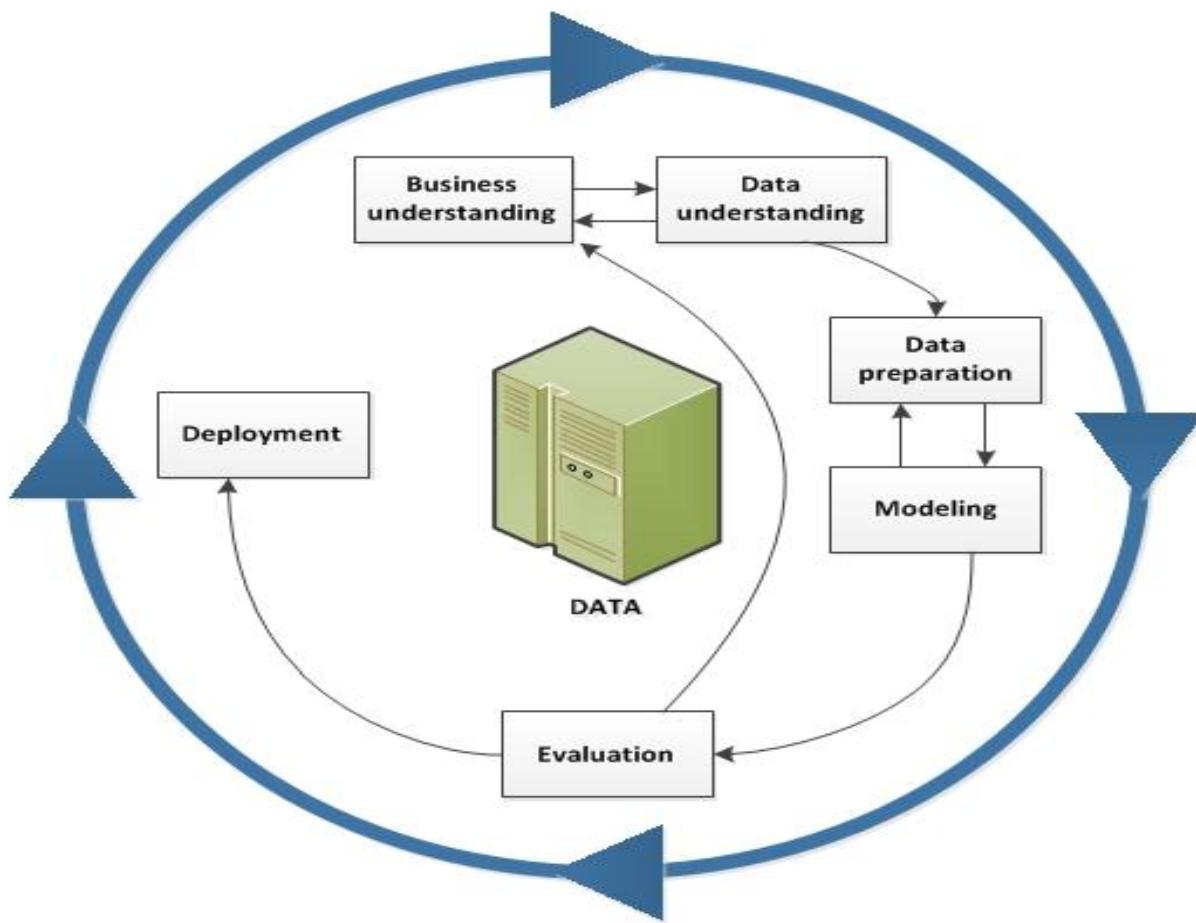


Fig : Data science process flowchart

Ref : [https://en.wikipedia.org/wiki/Data\\_analysis](https://en.wikipedia.org/wiki/Data_analysis)

The data analytics encompasses six phases that are data discovery, data aggregation, planning of the data models, data model execution, communication of the results, and operationalization.



## Fig: Data Analysis Process

Ref : <https://www.ibm.com/support/knowledgecenter/>

Let's understand the basic phases in Data Science Project

## 1. Business Understanding

- Data mining goals are defined.
  - The fundamental requirement is to understand client and business objectives.
  - Current data mining scenario factors in resources, constraints and assumptions should also be taken into the consideration.

## 2. Data Understanding

- In this stage, a sanity check is conducted to understand whether it is appropriate for data mining goals.
  - The data is collected from various sources within the organization.
  - It is a highly complex process since data and process from various sources are unlikely to match easily.

### 3. Data Preparation

- The data is production ready in this stage.

- The data from diverse sources should be nominated, cleaned, transformed, formatted, anonymized, and created.
- Data cleaning is a process to "clean" the data by smoothing noisy data and satisfying in missing values.

#### 4. Modelling

- In this stage, mathematical models are used to determine the data patterns.
- Suitable modelling techniques need to be chosen for the prepared data set.
- After that, create a scenario to validate the model. Then run the model on the prepared data set.

#### 5. Evaluation

- In this stage, patterns recognized are examined against business objectives.
- A go or no-go decision should be taken to move the model in the deployment phase.

#### 6. Deployment

- In this stage, ship your data mining.
- Discoveries (model/reports) to every business operation.
- A thorough deployment plan, for shipping, maintenance, and monitoring of data mining discoveries is created. – Knowledge Management.

Let us understand the difference between Descriptive and Predictive analytics

#### Descriptive vs Predictive Analytics

- **Descriptive Analytics:** Describing or summarising the existing data using existing business intelligence tools to better understand what is going on or what has happened.
- **Predictive Analytics:** Emphasizes on predicting the possible outcome using statistical models and machine learning techniques.

Basis for Comparison	Descriptive Analytics	Predictive Analytics
Describes	What happened in the past? By using the stored data.	What might happen in the future? By using the past data and analyzing it.
Process Involved	Involves Data Aggregation and Data Mining.	Involves Statistics and forecast techniques.
Definition	The process of finding useful and important information by analyzing the huge data.	This process involves in forecasting the future of the company, which are very useful.
Data Volume	It involves in processing huge data that are stored in data warehouses. Limited to past data.	It involves analyzing large past data and then predicts the future using advance techniques.
Examples	Sales report, revenue of a company, performance analysis, etc.	Sentimental analysis, credit score analysis, forecast reports for a company, etc.
Accuracy	It provides accurate data in the reports using past data.	Results are not accurate, it will not tell you exactly what will happen but it will tell you what might happen in the future.
Approach	It allows the reactive approach	While this a proactive approach

## Descriptive Analytics

These describe what has already happened.

- For example, a retailer can learn the number of clients, the average bill, the most popular goods, etc.
- A medical company can evaluate the most common illnesses and susceptibility to disease.

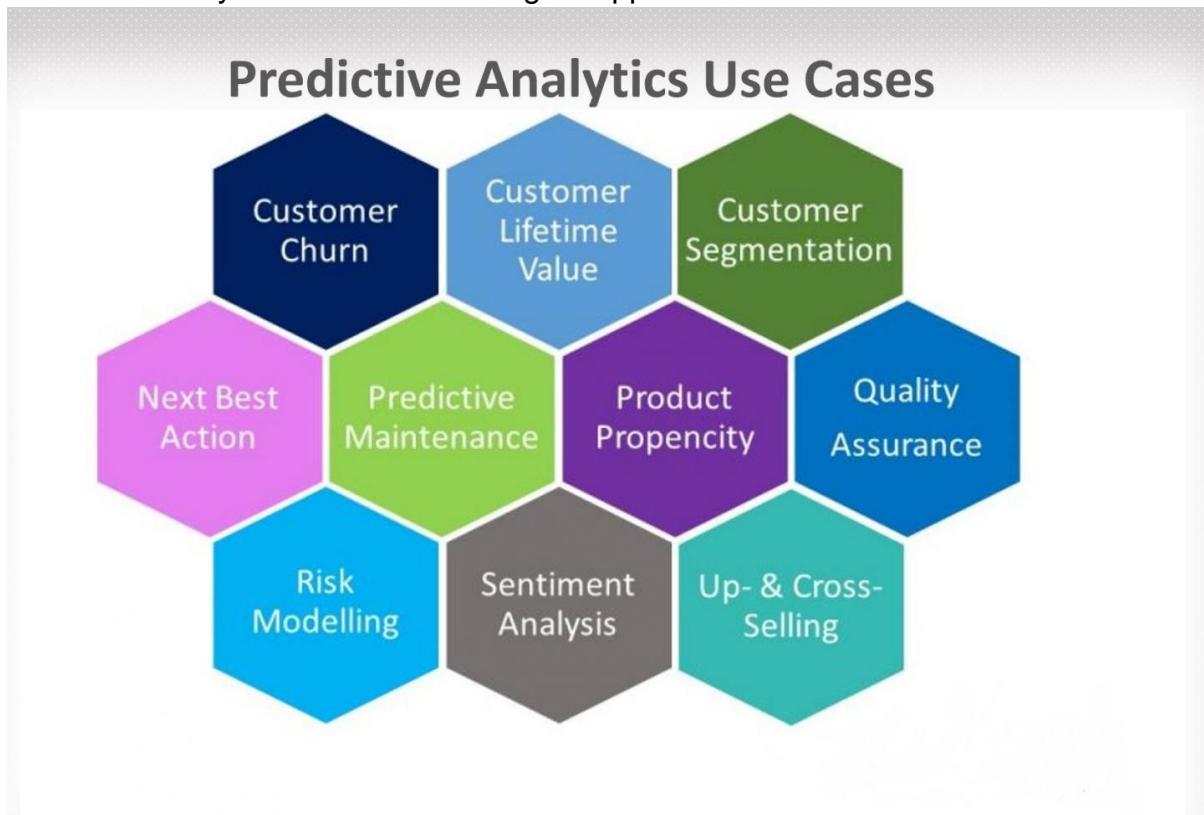
- Governments can stimulate population growth by organising additional social support for families with several children if the overall number of pregnant women has been decreasing over several consistent years.

With the help of descriptive analysis, any company is able to group its customers by social factors, behaviour and other features, as well as monitoring peak activities according to seasonal or local factors.

Descriptive analytics manipulates figures from multiple data sources to provide valuable information about the past. So, developers can predict important trends and signal the necessity of preventative or stimulative actions.

## Predictive analytics

Predictive analytics address what might happen.



To be able to predict trends and see into the future, this type of analytics uses the results of the previous two – i.e. it bases its results on true facts of the past.

With the help of the predictive analysis, an entrepreneur can optimise the raw material storage and the warehouse stock. Computer systems predict stock exchanges, market fluctuations and currency exchange rates. This is specifically useful in finance, production, logistics and banking.

However, it's important to understand that all the results this type of analytics provides you with are approximate. The accuracy of data and the stability of the situation have a significant influence on the result. It requires careful processing and constant optimisation.

## Tools for Data Analysis

Many tools are available in the market, which make it easier for us :

- 1) To process and manipulate data
- 2) Analyze the relationships and correlations between data sets,
- 3) Helps to identify patterns and trends for interpretation
  - Python
  - Tableau
  - R Programming
  - Power BI

## 2.8 Anaconda Software and Introduction to Python

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 as 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 is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.

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.

Navigator is a desktop graphical user interface that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands.

## Anaconda Navigator

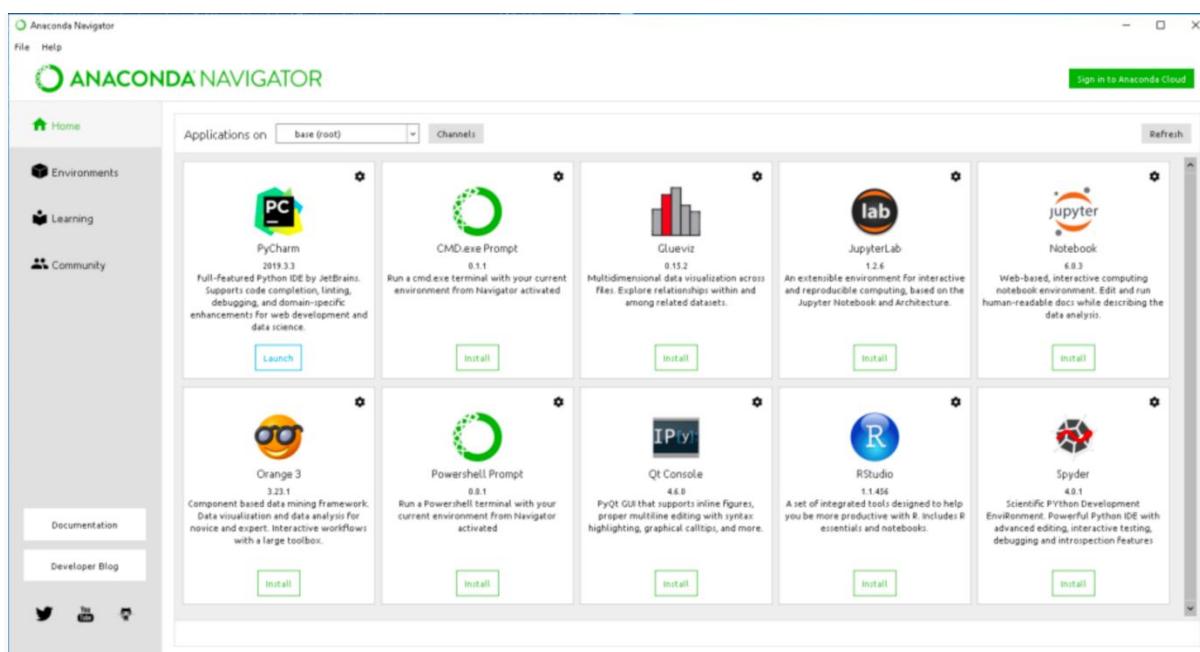


Fig : Anaconda Navigator Interface

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator;

- [JupyterLab](#)
- [Jupyter Notebook](#)
- [QtConsole](#)
- [Spyder](#)
- [Glue](#)
- [Orange](#)
- [RStudio](#)
- [Visual Studio Code](#)

## Conda

Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs, and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi-language projects. The conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

## Anaconda Cloud

Anaconda Cloud is a package management service by Anaconda where users can find, access, store and share public and private notebooks, environments, and conda and PyPI packages. Cloud hosts useful Python packages, notebooks and environments for a wide variety of applications. Users do not need to log in or to have a Cloud account, to search for public packages, download and install them. Users can build new packages using the Anaconda Client command line interface (CLI), then manually or automatically upload the packages to Cloud.

Now let's get started with very popular programming language used in data science and variety of other tasks like website building, server-side programming etc. Named Python.

## Introduction to 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?*

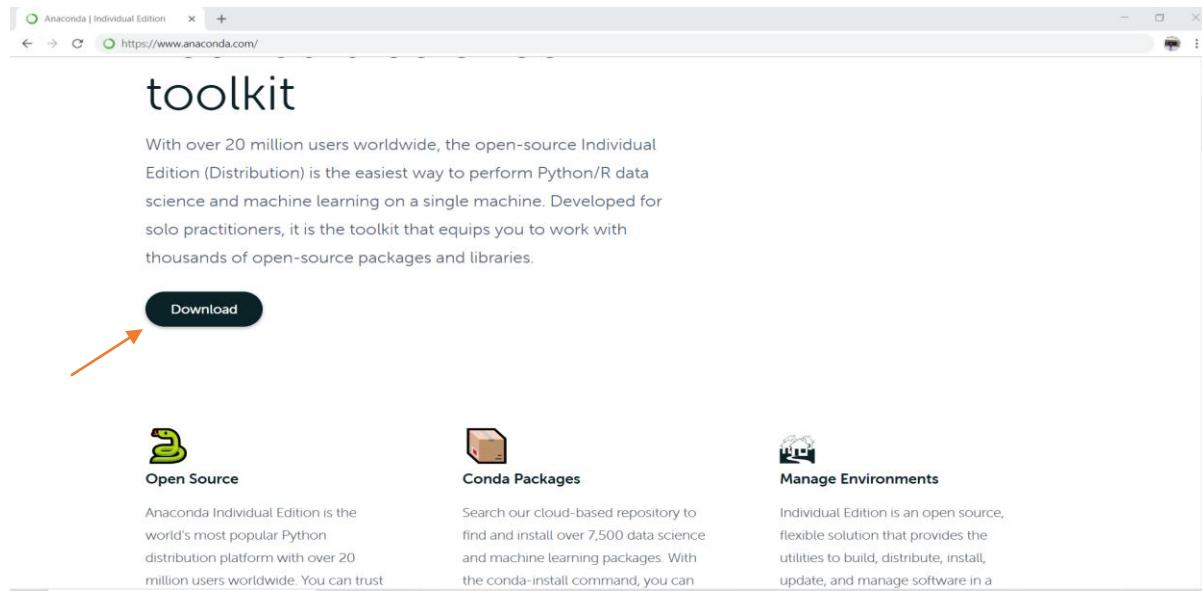
- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax like the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning 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.

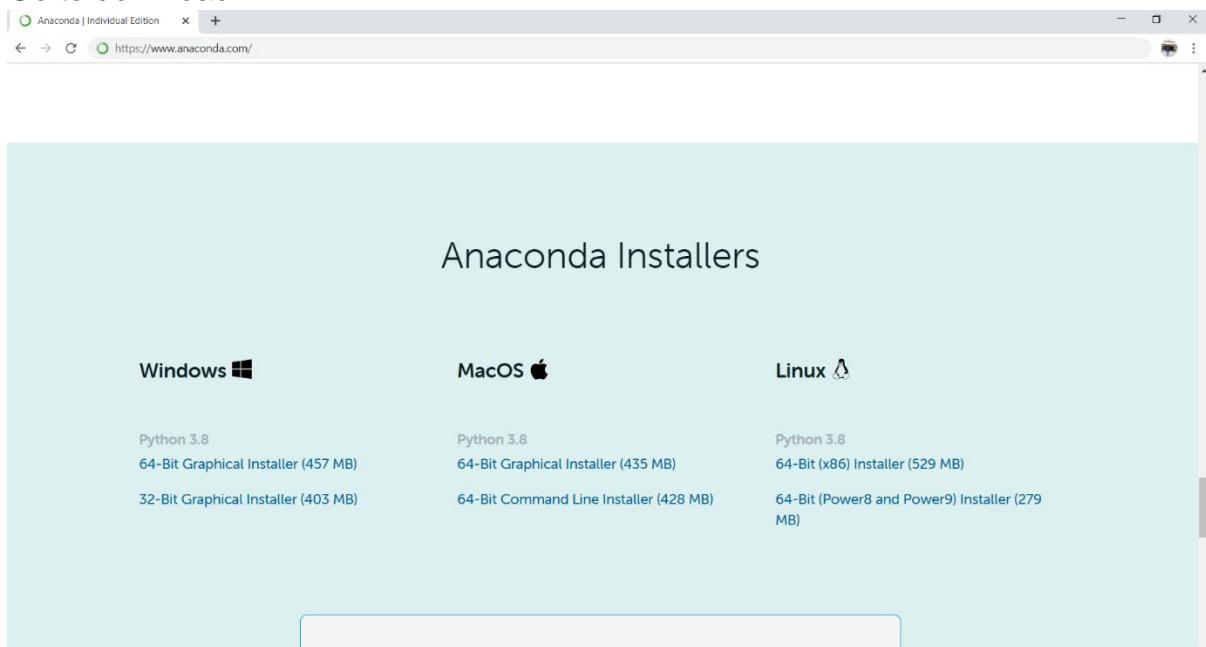
## 2.9 Installing Anaconda

1. Go to [anaconda.com](http://anaconda.com)



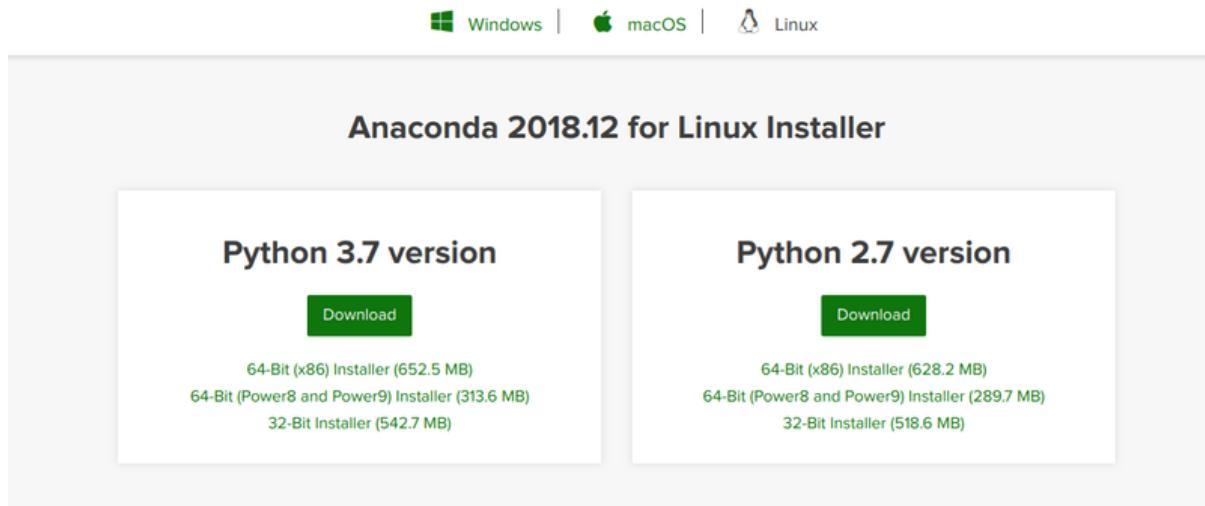
The screenshot shows the Anaconda Individual Edition website. At the top, there's a header with the SAP logo and the word "toolkit". Below the header, there's a paragraph of text about the toolkit, followed by a large "Download" button. An orange arrow points to this "Download" button. To the left of the download button is a section titled "Open Source" with a Python logo icon. To the right of the download button are sections for "Conda Packages" (with a Conda logo icon) and "Manage Environments" (with a server logo icon).

## 2. Go to download.

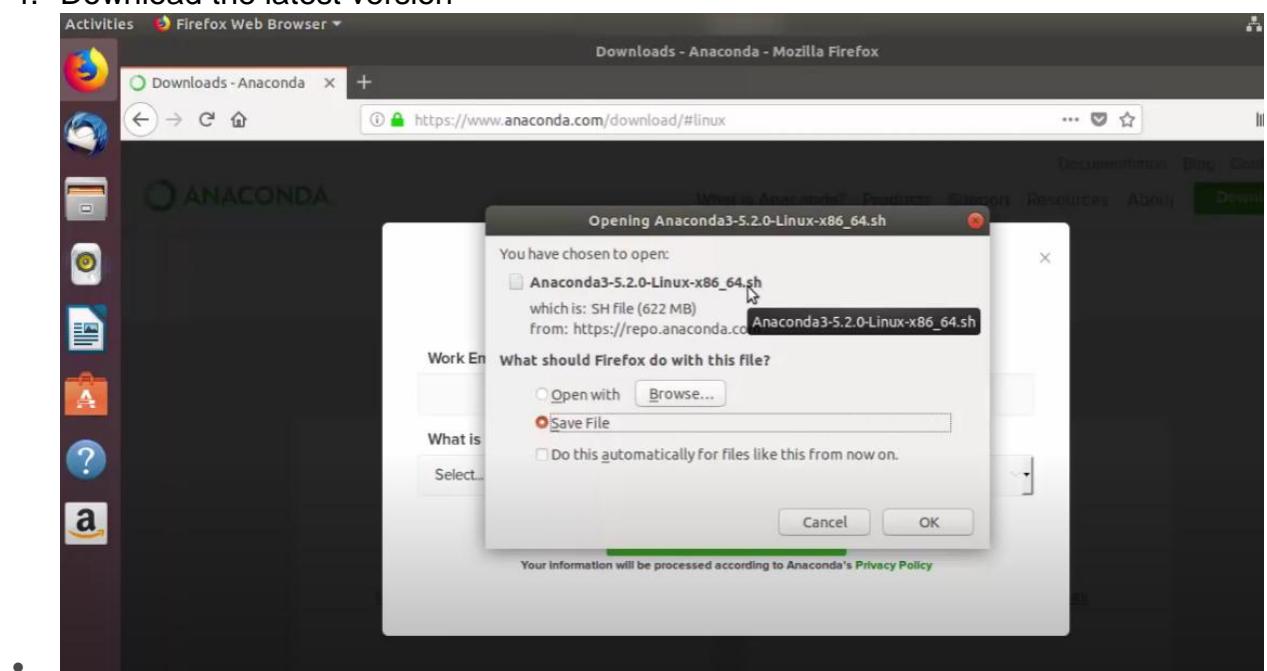


The screenshot shows the "Anaconda Installers" page. It features three main download options: "Windows" (with a Windows logo icon), "MacOS" (with a Mac logo icon), and "Linux" (with a Linux logo icon). Each option has two download links: "Python 3.8" and "64-Bit Graphical Installer". The "Windows" section also includes a link for "32-Bit Graphical Installer". The "MacOS" section includes a link for "64-Bit Command Line Installer". The "Linux" section includes a link for "64-Bit (Power8 and Power9) Installer".

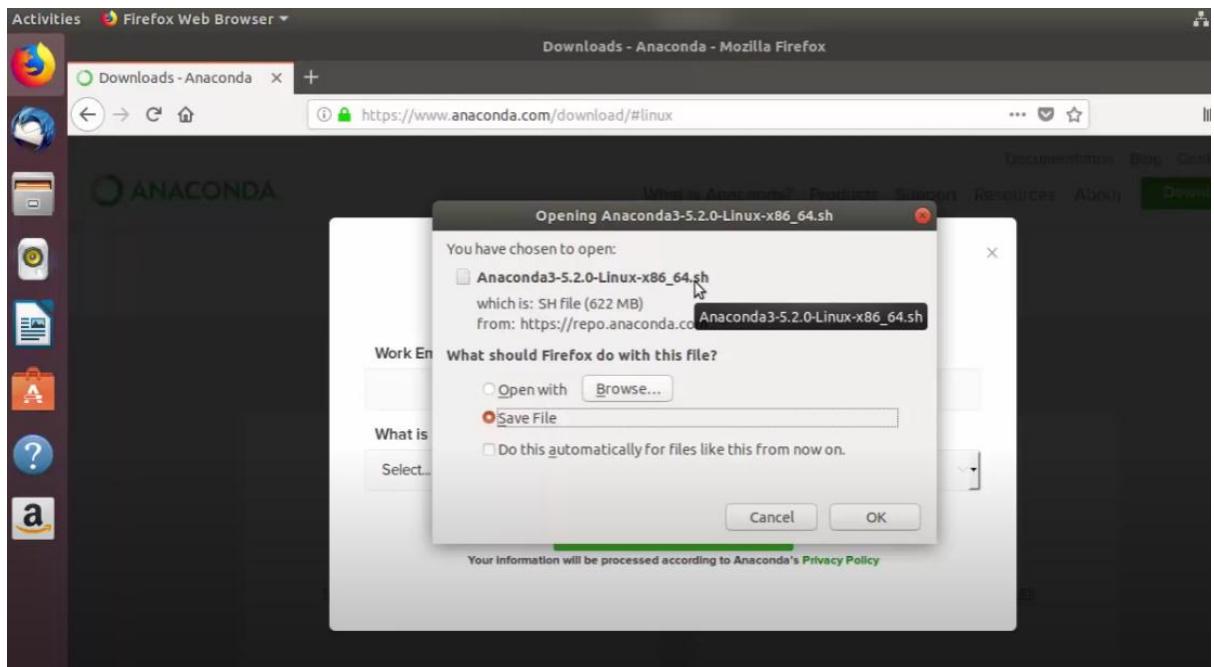
## 3. Choose your version



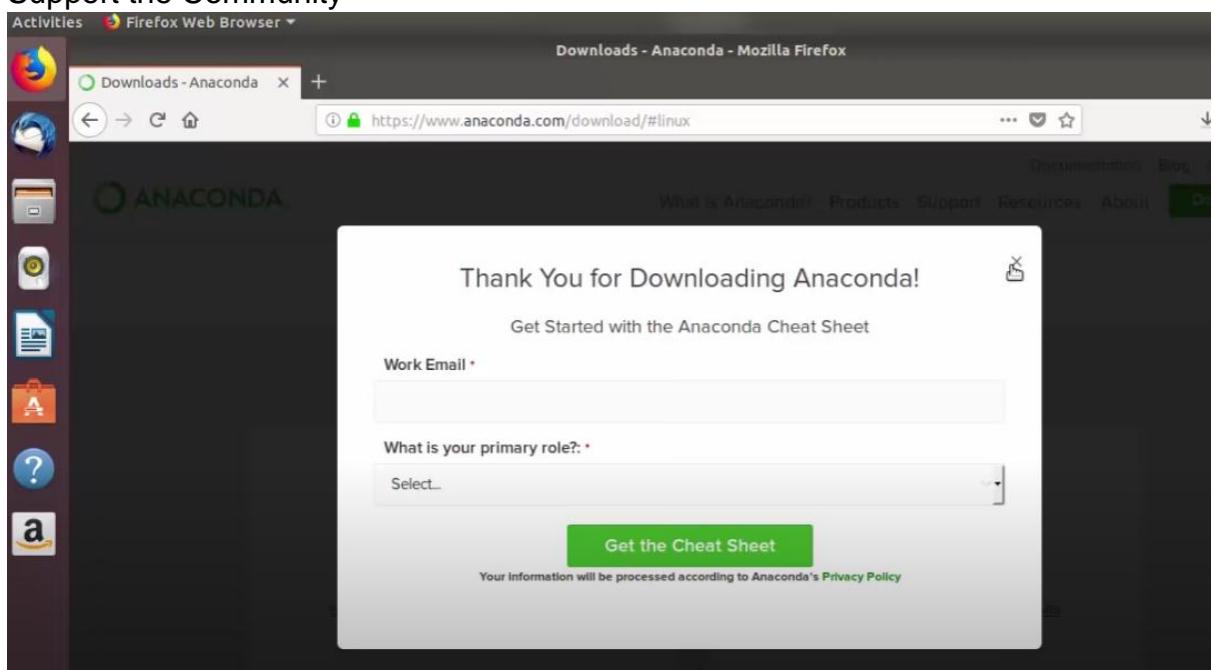
#### 4. Download the latest version



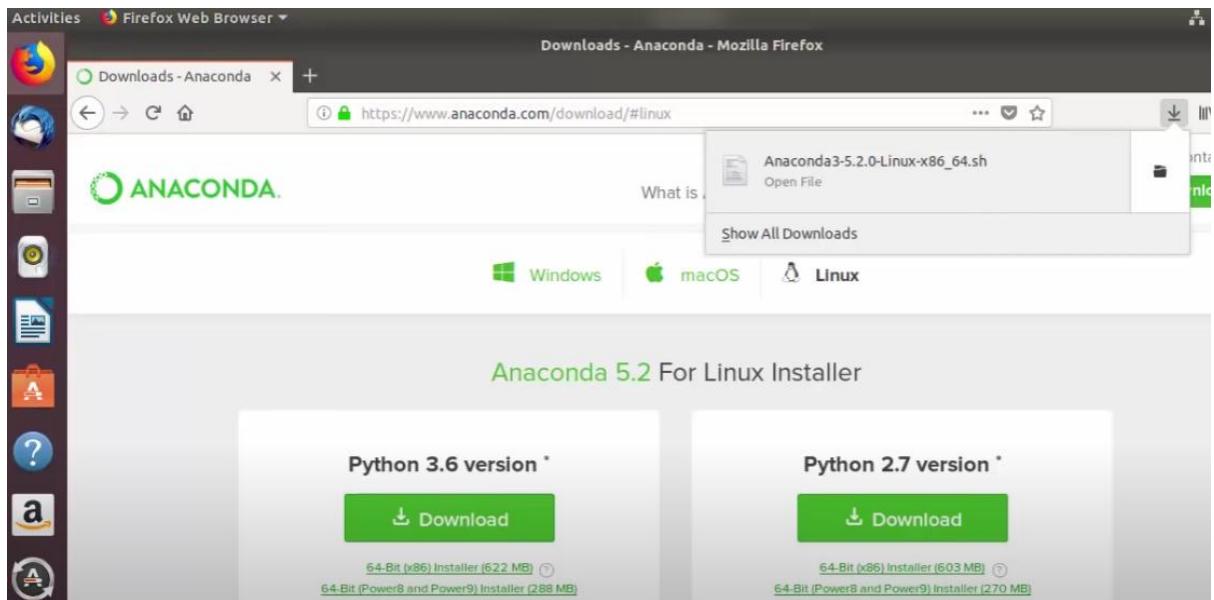
#### 5. Save



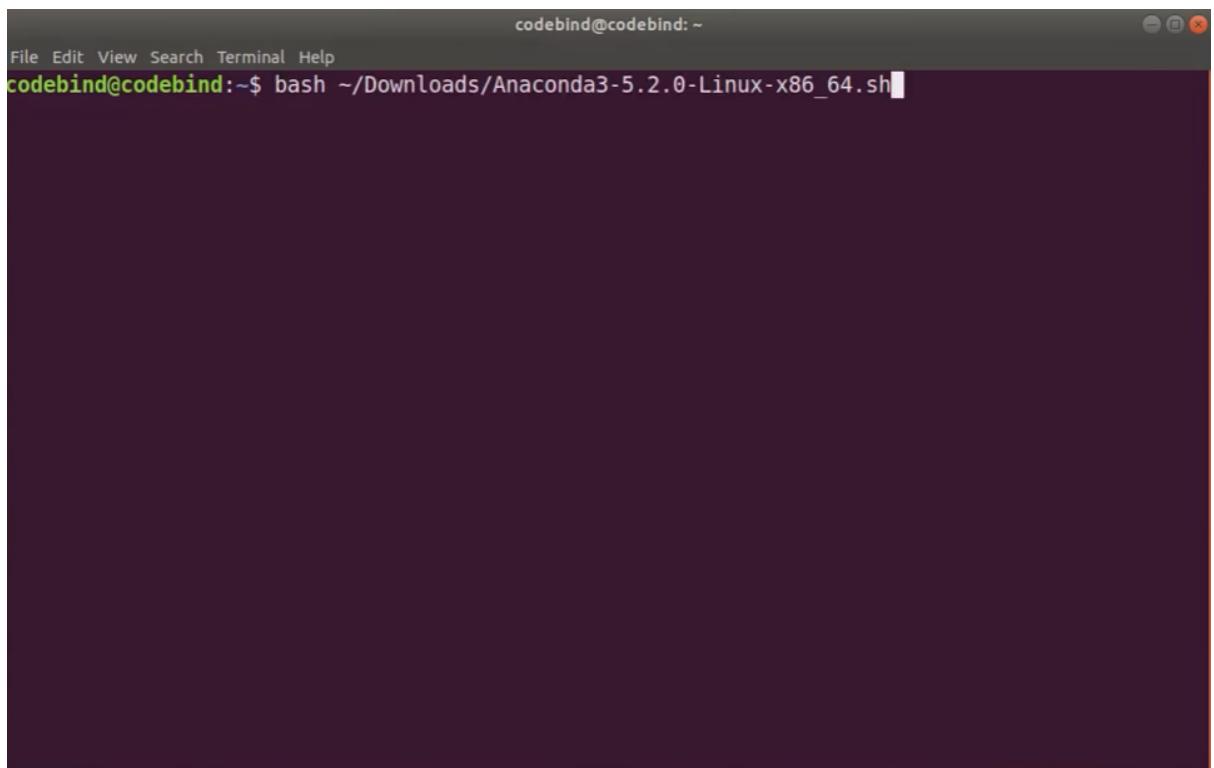
## 6. Support the Community



## 7. Run the file from download folder.



8. Open Terminal & run the following command



```
codebind@codebind: ~
File Edit View Search Terminal Help
codebind@codebind:~$ bash ~/Downloads/Anaconda3-5.2.0-Linux-x86_64.sh
```

9. Let the process continue

```
codebind@codebind:~  
File Edit View Search Terminal Help  
codebind@codebind:~$ bash ~/Downloads/Anaconda3-5.2.0-Linux-x86_64.sh  
Welcome to Anaconda3 5.2.0  
In order to continue the installation process, please review the license  
agreement.  
Please, press ENTER to continue  
>>>
```

## 10. Accept the terms

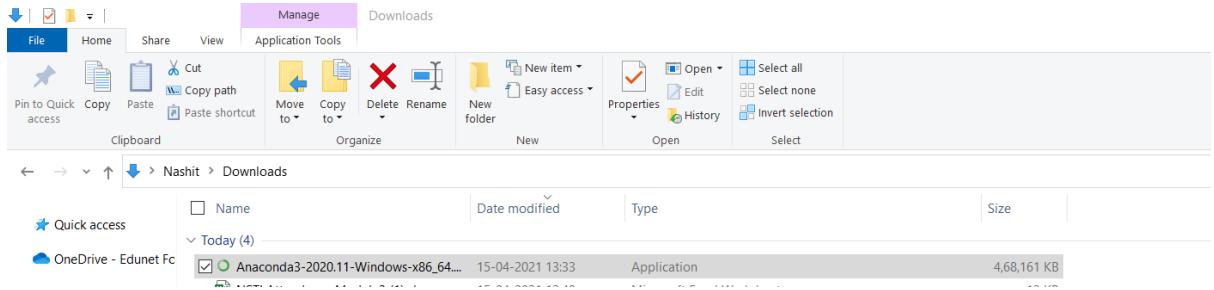
```
=====  
Copyright 2015, Anaconda, Inc.  
All rights reserved under the 3-clause BSD License:  
  
Redistribution and use in source and binary forms, with or without modification,  
are permitted provided that the following conditions are met:  
  
    * Redistributions of source code must retain the above copyright notice, this  
list of conditions and the following disclaimer.  
    * Redistributions in binary form must reproduce the above copyright notice, th  
is list of conditions and the following disclaimer in the documentation and/or o  
ther materials provided with the distribution.  
    * Neither the name of Anaconda, Inc. ("Anaconda, Inc.") nor the names of its c  
ontributors may be used to endorse or promote products derived from this softwar  
e without specific prior written permission.  
  
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WA  
RRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
  
Do you accept the license terms? [yes|no]  
[no] >>> yes
```

## 11. Anaconda is now installed.

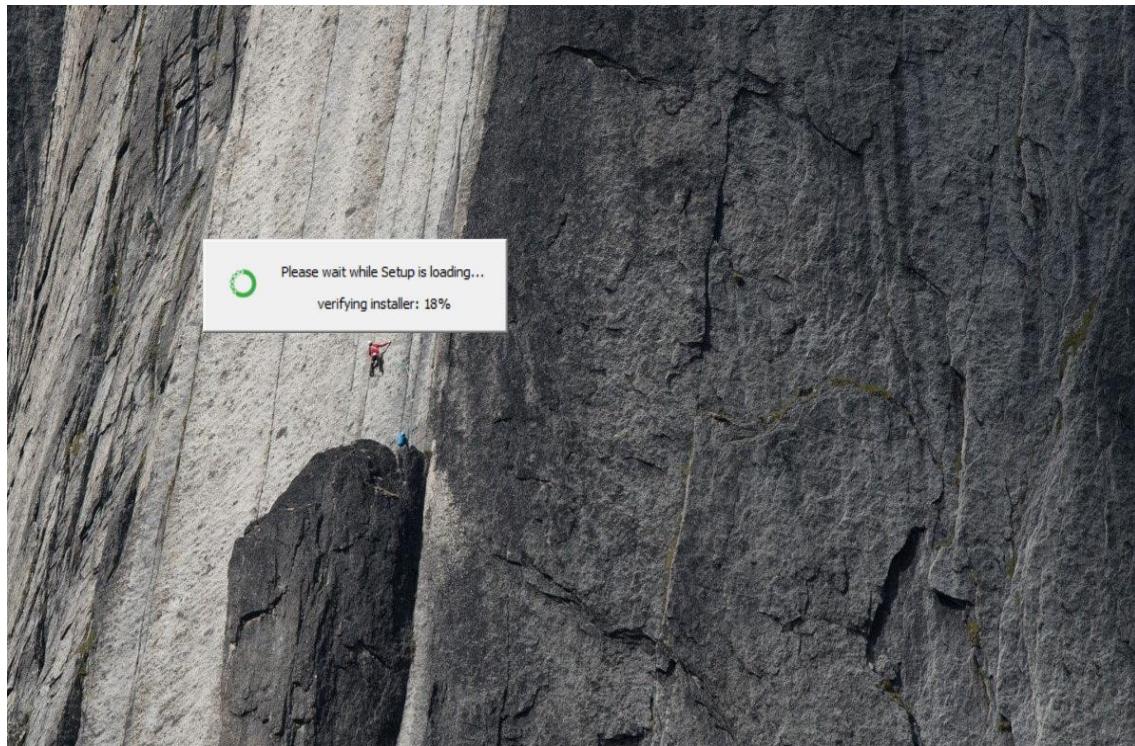
## For Windows

Just follows the following steps :

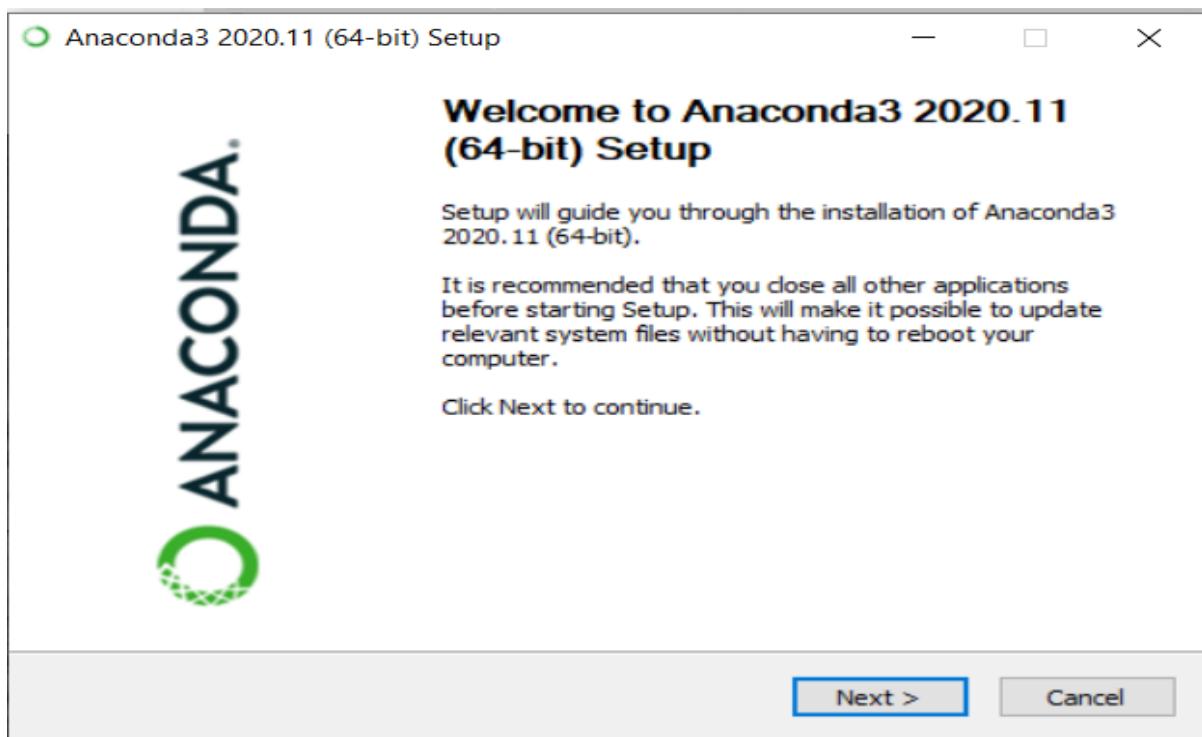
1. Locate the downloaded file



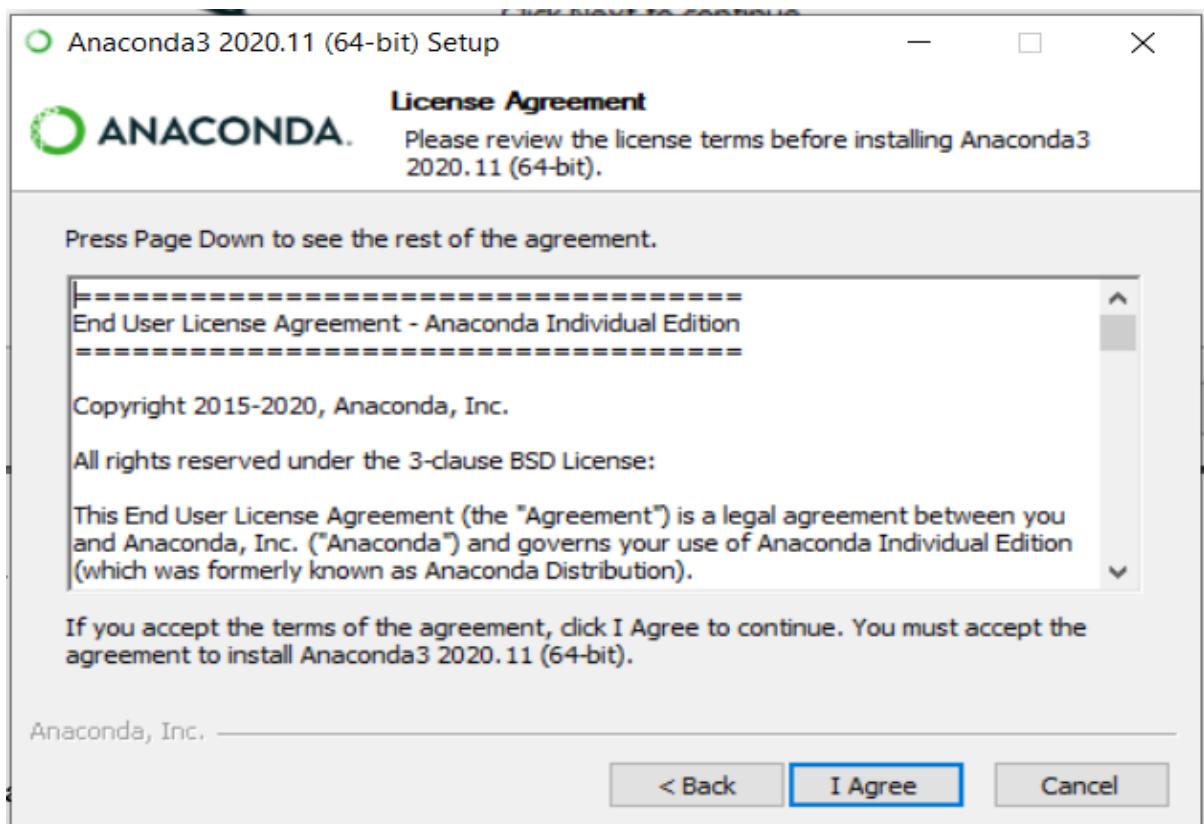
2. Run the setup.



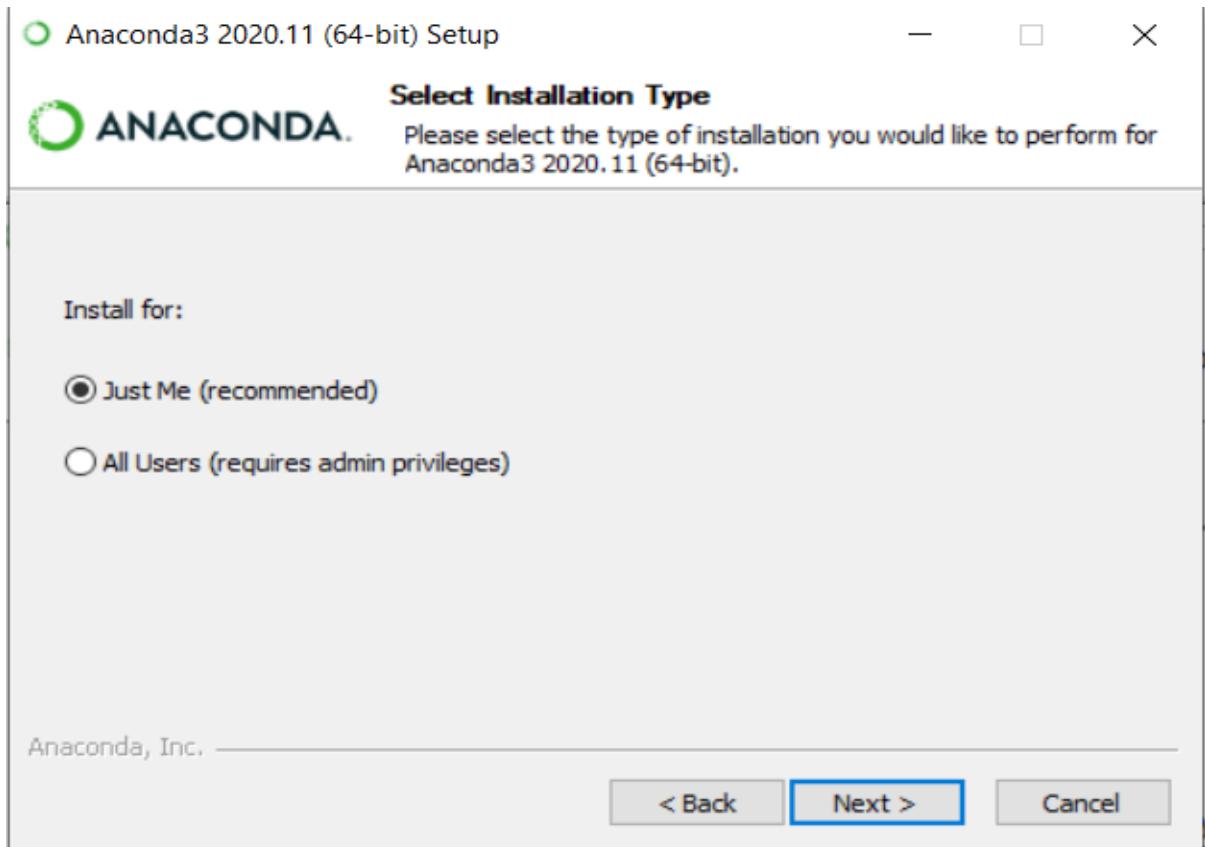
3. Next



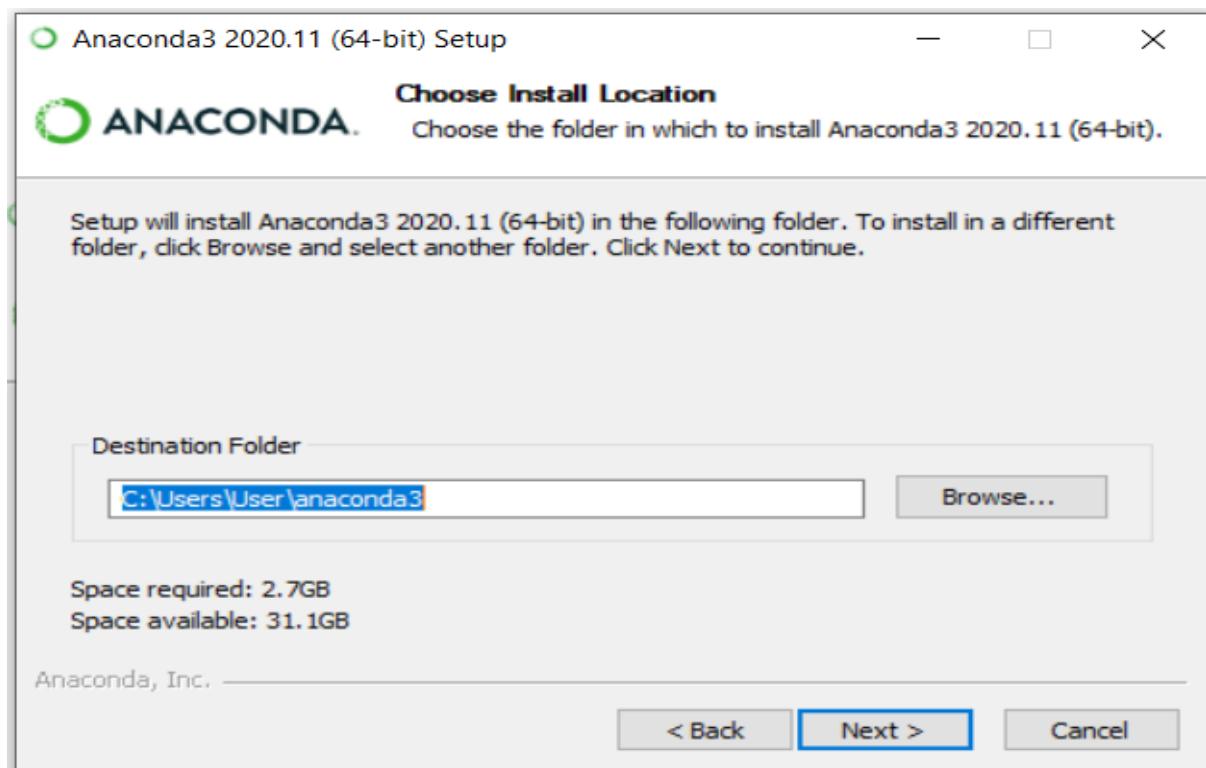
4. Agree to the Terms.



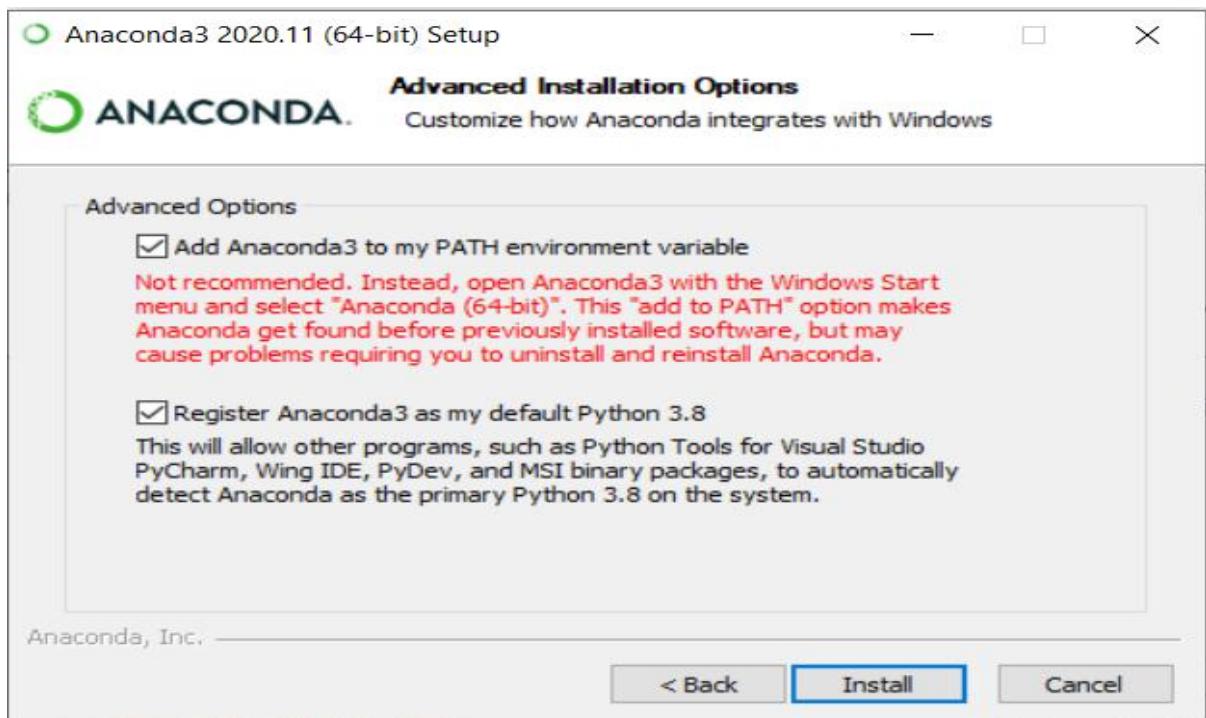
5. For beginners, or individual users “Just me”.



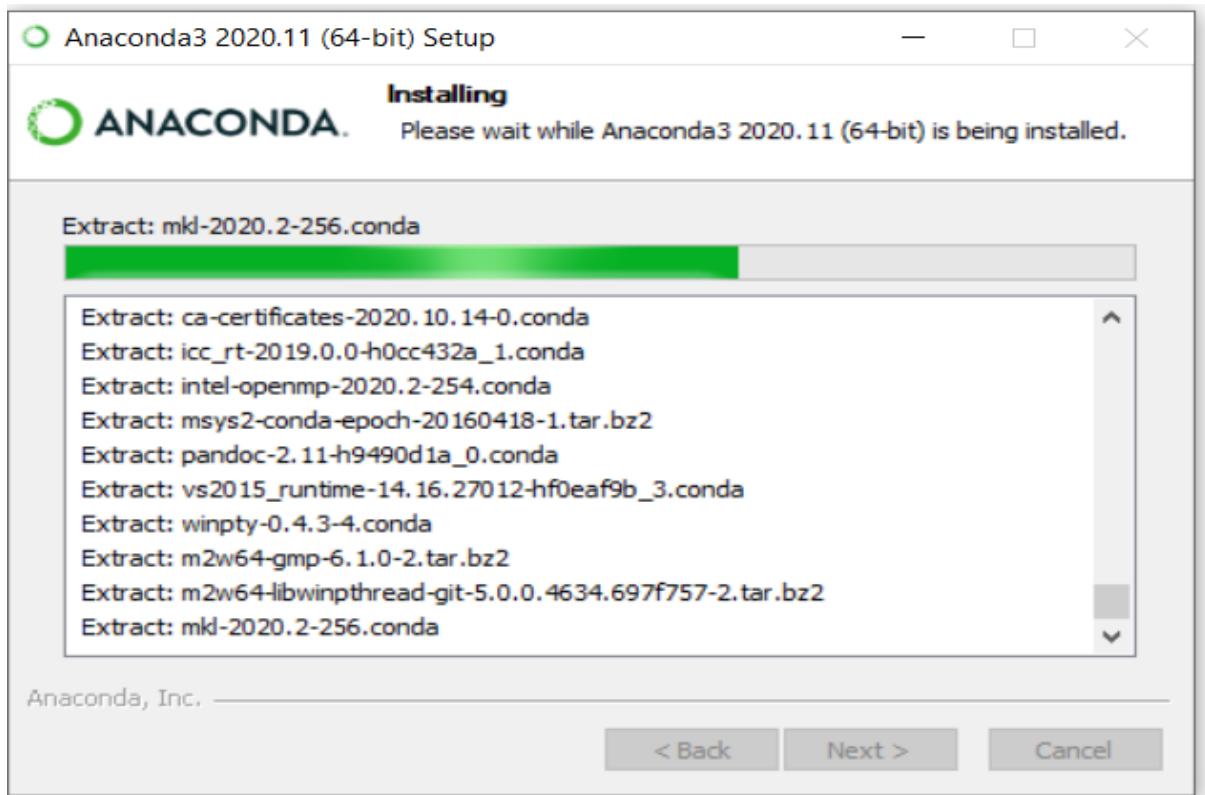
6. Browse the location, where you want to install. Keep it default, & Next.



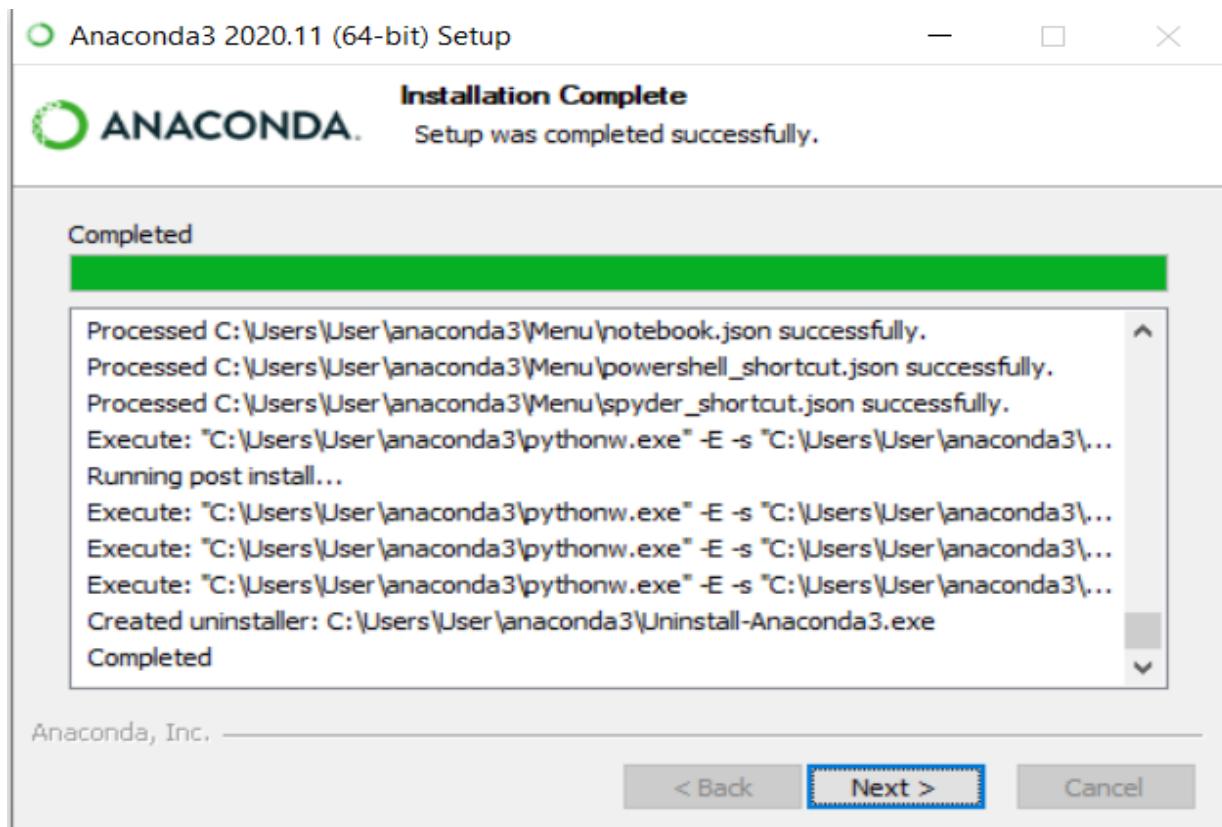
7. Let it be added to PATH Environment Variables, to make it readily available through command prompt.



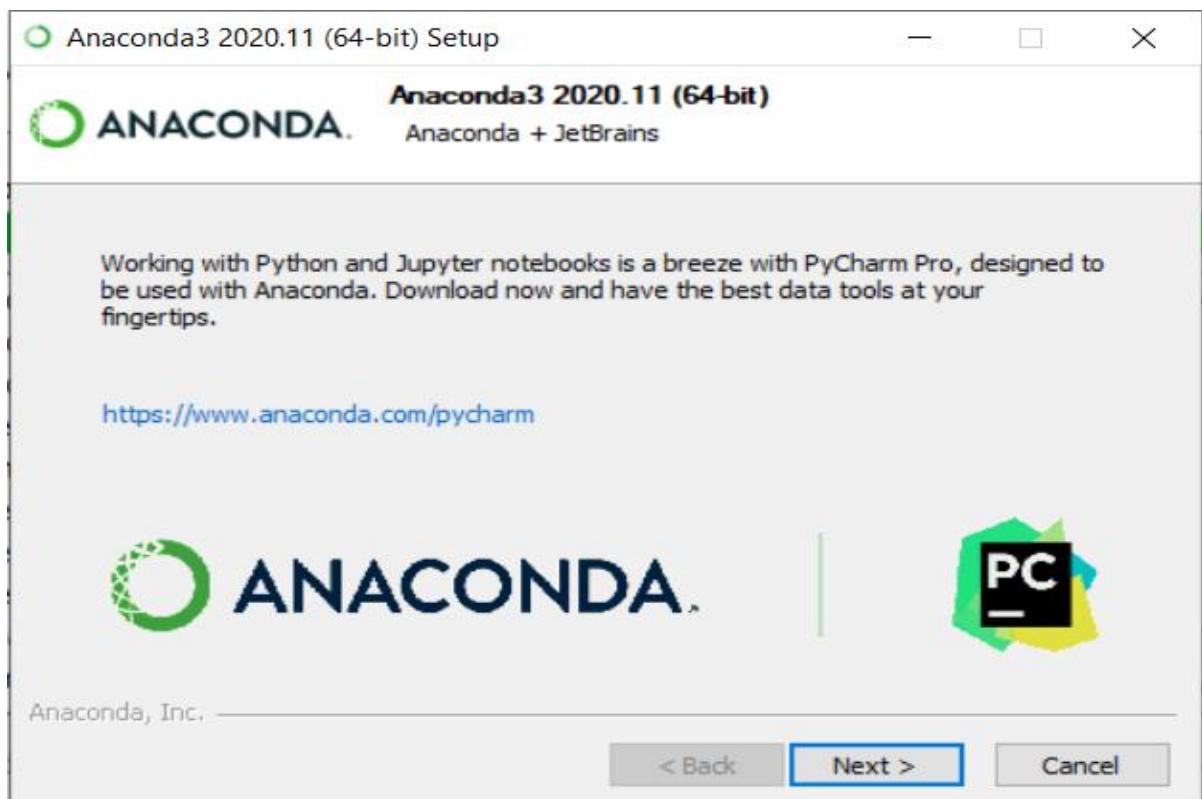
8. Install.
9. The installation process may take a few minutes.



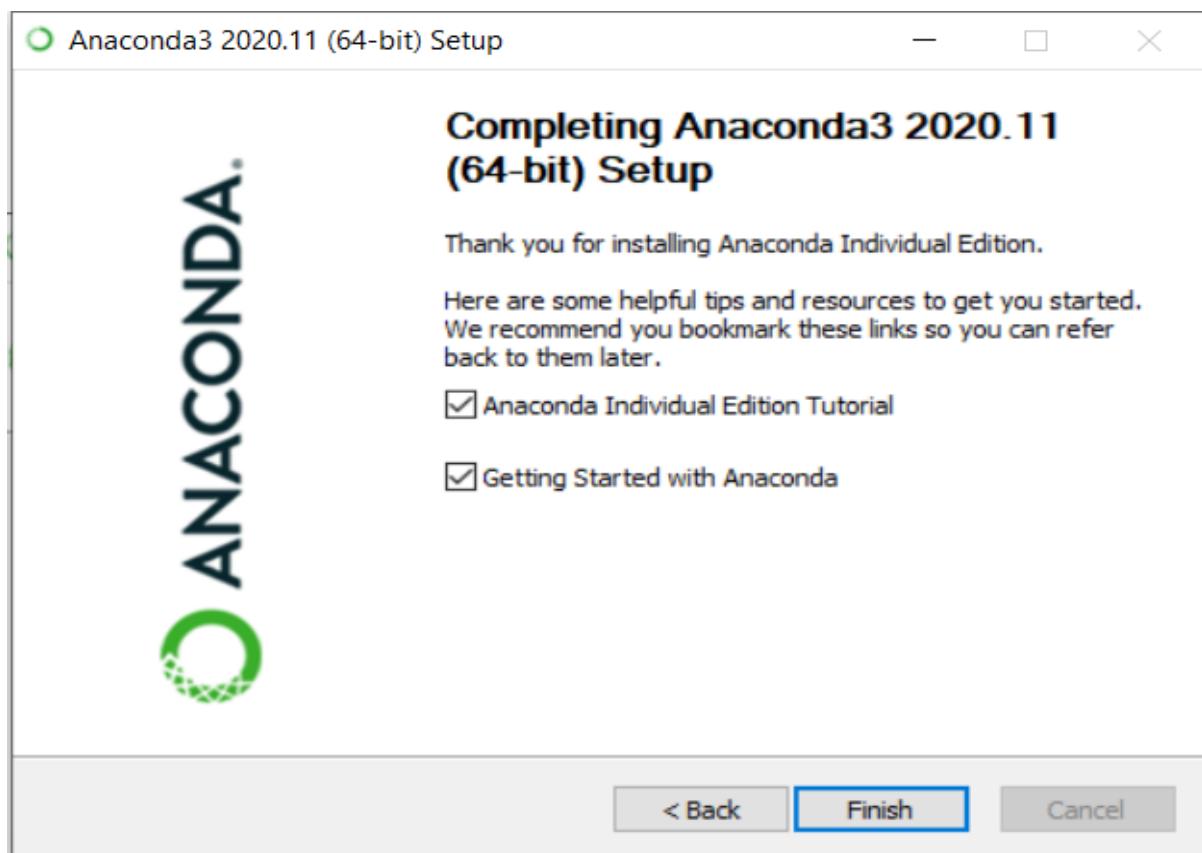
10. Once installation is complete, press next.



## 11. Other packages.



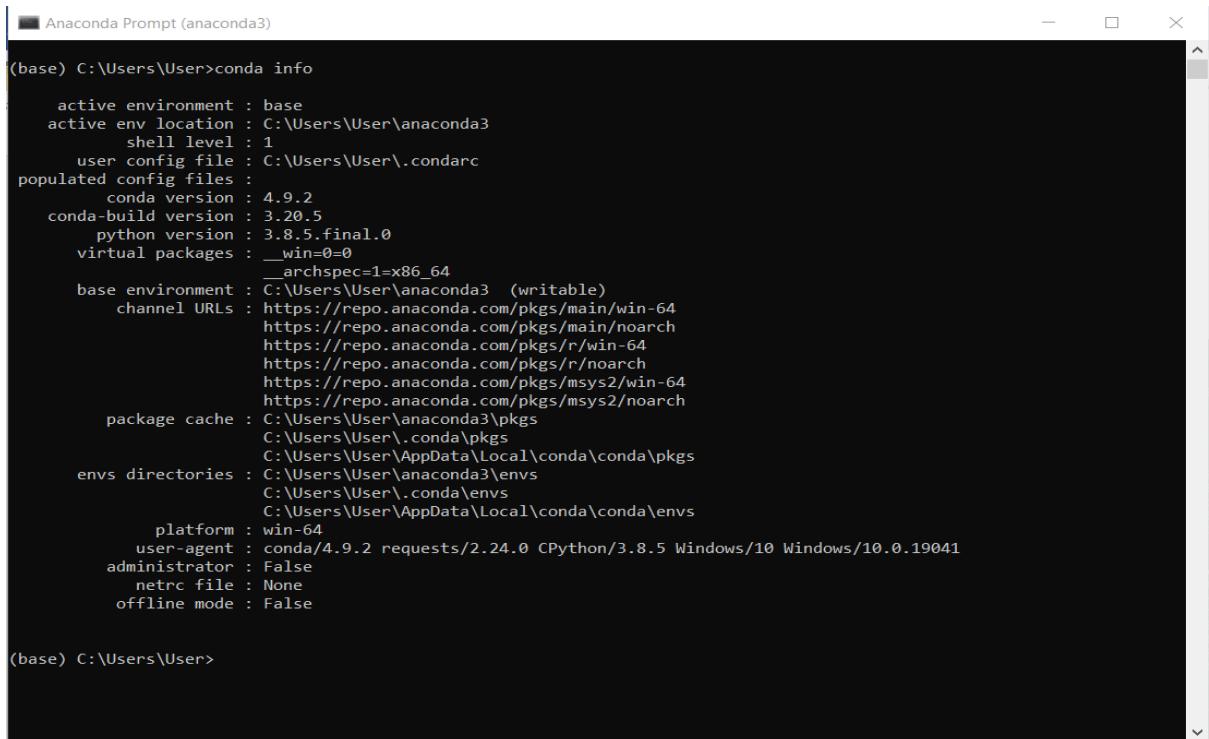
## 12. Finish Setup



## Anaconda Prompt

1. Start Anaconda prompt from Start Menu

2. Use command to check out your conda,  
conda info



```
(base) C:\Users\User>conda info

active environment : base
active env location : C:\Users\User\anaconda3
    shell level : 1
    user config file : C:\Users\User\.condarc
populated config files :
    conda version : 4.9.2
    conda-build version : 3.20.5
    python version : 3.8.5.final.0
    virtual packages : __win=0=0
                        __archspec=1=x86_64
base environment : C:\Users\User\anaconda3 (writable)
    channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                    https://repo.anaconda.com/pkgs/main/noarch
                    https://repo.anaconda.com/pkgs/r/win-64
                    https://repo.anaconda.com/pkgs/r/noarch
                    https://repo.anaconda.com/pkgs/msys2/win-64
                    https://repo.anaconda.com/pkgs/msys2/noarch
    package cache : C:\Users\User\anaconda3\pkgs
                    C:\Users\User\.conda\pkgs
                    C:\Users\User\AppData\Local\conda\conda\pkgs
envs directories : C:\Users\User\anaconda3\envs
                    C:\Users\User\.conda\envs
                    C:\Users\User\AppData\Local\conda\conda\envs
    platform : win-64
    user-agent : conda/4.9.2 requests/2.24.0 CPython/3.8.5 Windows/10 Windows/10.0.19041
    administrator : False
    netrc file : None
    offline mode : False

(base) C:\Users\User>
```

3. Check the Environment available;  
conda info –envs  
conda env list

```
Anaconda Prompt (anaconda3) - conda create --name myenv
conda-build version : 3.20.5
python version : 3.8.5.final.0
virtual packages : __win=0=0
                   __archspec=1=x86_64
base environment : C:\Users\User\anaconda3 (writable)
channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                 https://repo.anaconda.com/pkgs/main/noarch
                 https://repo.anaconda.com/pkgs/r/win-64
                 https://repo.anaconda.com/pkgs/r/noarch
                 https://repo.anaconda.com/pkgs/msys2/win-64
                 https://repo.anaconda.com/pkgs/msys2/noarch
package cache : C:\Users\User\anaconda3\pkgs
                 C:\Users\User\.conda\pkgs
                 C:\Users\User\AppData\Local\conda\conda\pkgs
envs directories : C:\Users\User\anaconda3\envs
                   C:\Users\User\.conda\envs
                   C:\Users\User\AppData\Local\conda\conda\envs
platform : win-64
user-agent : conda/4.9.2 requests/2.24.0 CPython/3.8.5 Windows/10 Windows/10.0.19041
administrator : False
netrc file : None
offline mode : False

(base) C:\Users\User>conda info --envs
# conda environments:
#
base          *  C:\Users\User\anaconda3

(base) C:\Users\User>conda env list
# conda environments:
#
base          *  C:\Users\User\anaconda3

(base) C:\Users\User>
(base) C:\Users\User>conda create --name myenv
```

#### 4. Welcome to Anaconda Prompt

#### 5. Update the version, if needed

```
conda update -n base -c defaults conda
```

```
Anaconda Prompt (anaconda3)
(base) C:\Users\User>conda update -n base -c defaults conda
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\User\anaconda3

added / updated specs:
- conda

The following packages will be downloaded:

  package          |      build
-----|-----
backports.functools_lru_cache-1.6.3| pyhd3eb1b0_0      9 KB
backports.tempfile-1.0   | pyhd3eb1b0_1      11 KB
conda-4.10.1           | py38haa95532_1    2.9 MB
conda-package-handling-1.7.3| py38h8cc25b3_1    721 KB
-----|-----
                           Total:       3.6 MB

The following packages will be UPDATED:

backports.functoo~          1.6.1-py_0 --> 1.6.3-pyhd3eb1b0_0
conda                         4.9.2-py38haa95532_0 --> 4.10.1-py38haa95532_1
conda-package-han~           1.7.2-py38h76e460a_0 --> 1.7.3-py38h8cc25b3_1

The following packages will be DOWNGRADED:

backports.tempfile           1.0-py_1 --> 1.0-pyhd3eb1b0_1

Proceed ([y]/n)?

Downloading and Extracting Packages
backports.tempfile-1 | 11 KB  | #####| 100%
backports.functools_| 9 KB   | #####| 100%
conda-4.10.1         | 2.9 MB | #####| 100%
conda-package-handl| 721 KB | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\User>
(base) C:\Users\User>
```

## Anaconda Navigator

### Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

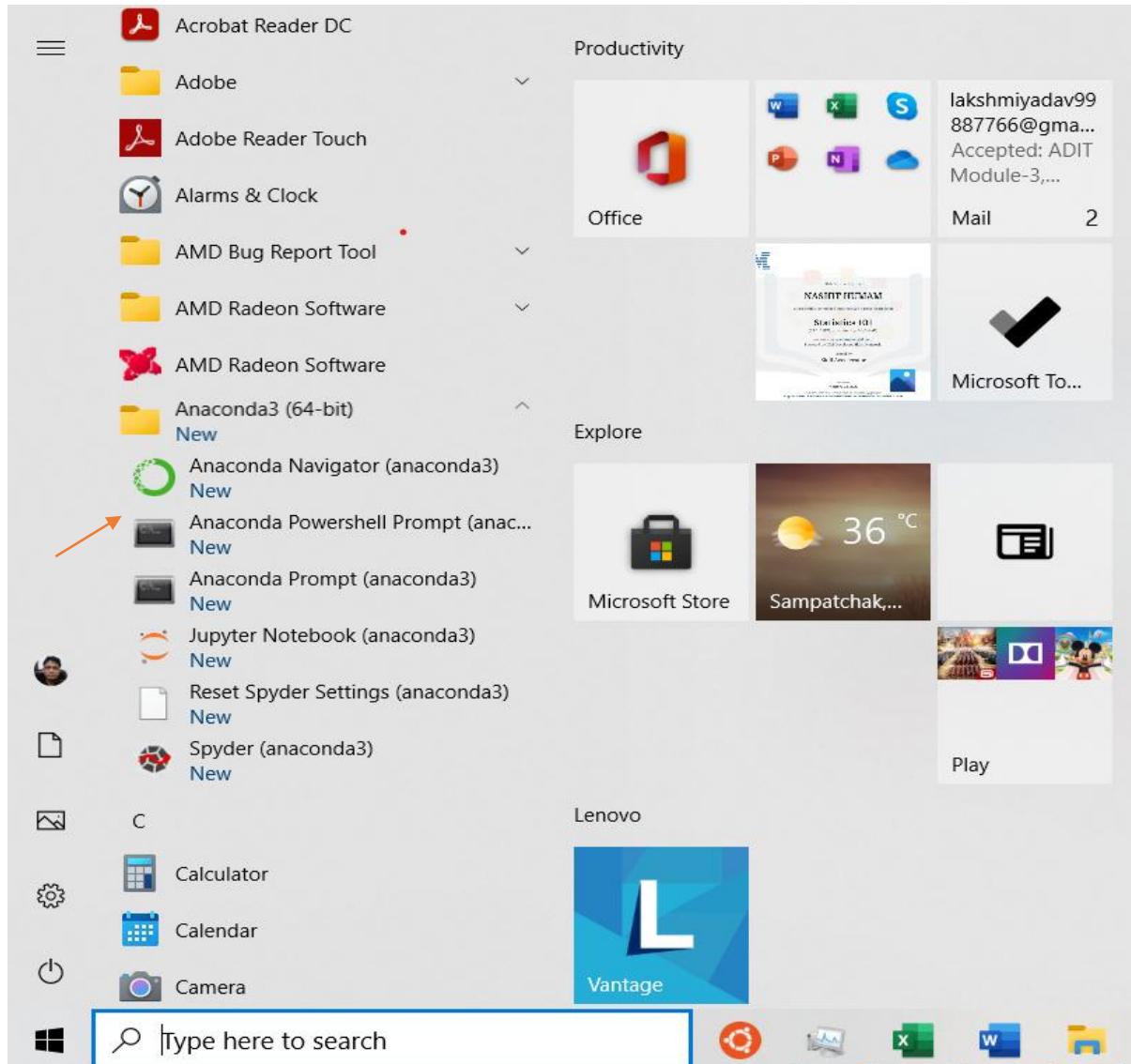
The command-line program `conda` is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type `conda` commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

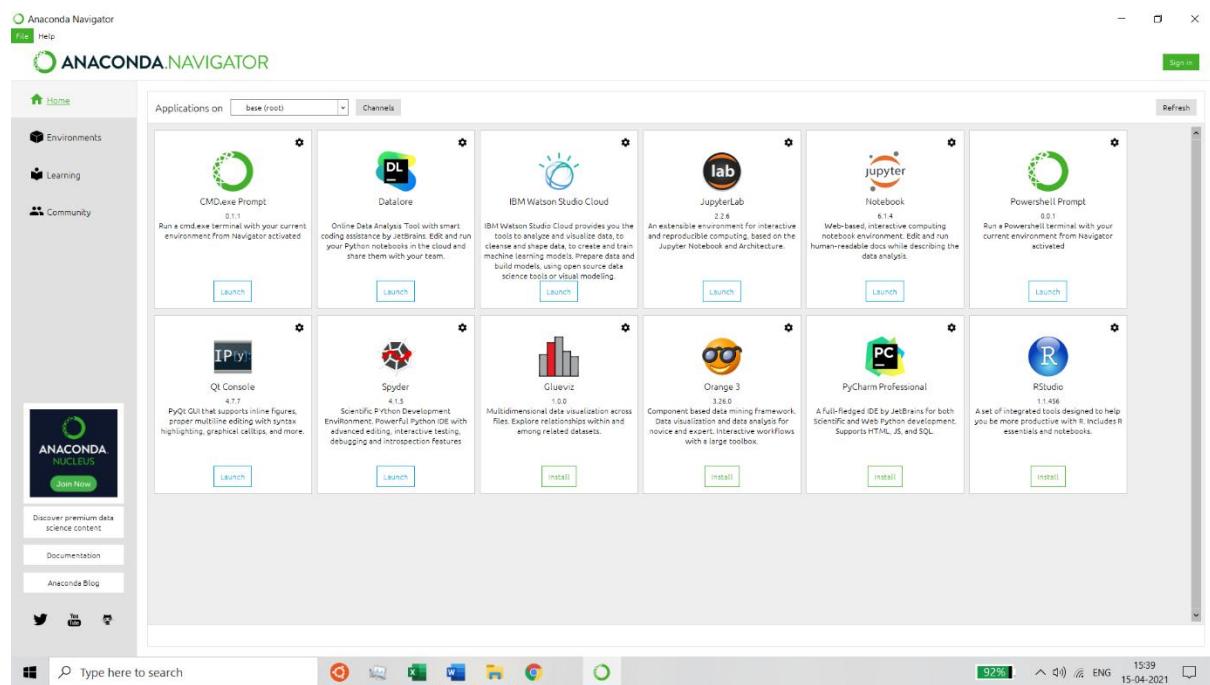
To get hold of Anaconda, we need to get used to both CLI, & GUI of the software.

We previously used the CLI version, now need to see the GUI version.

### 1. Start ANACONDA NAVIGATOR from Start Menu or Search menu



### 2. It provides you with the following GUI



### 3. Welcome to Anaconda Navigator.

You can launch the different packages from the Navigator.

#### Packages in Anaconda:

The following applications are available by default in Navigator:

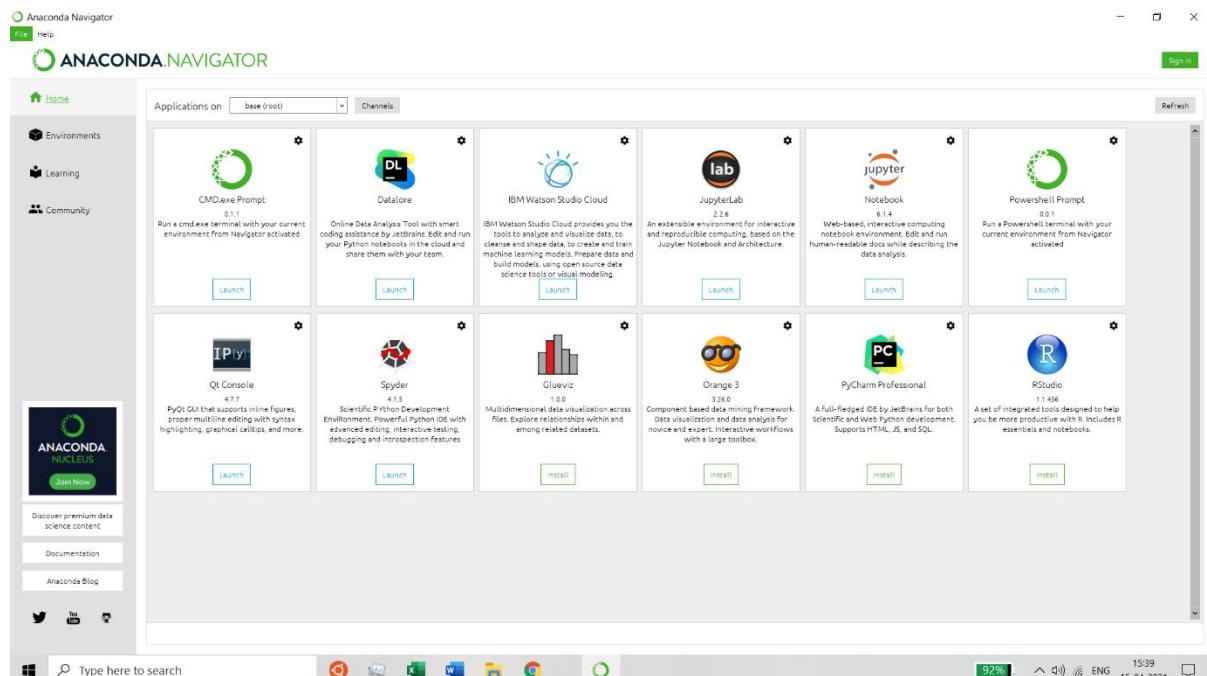
- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code

#### Jupyter Notebook

Jupyter is a free, open-source, interactive web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document. Or in easy words, IDE. The purpose is “to support interactive data science and scientific computing across all programming languages.”

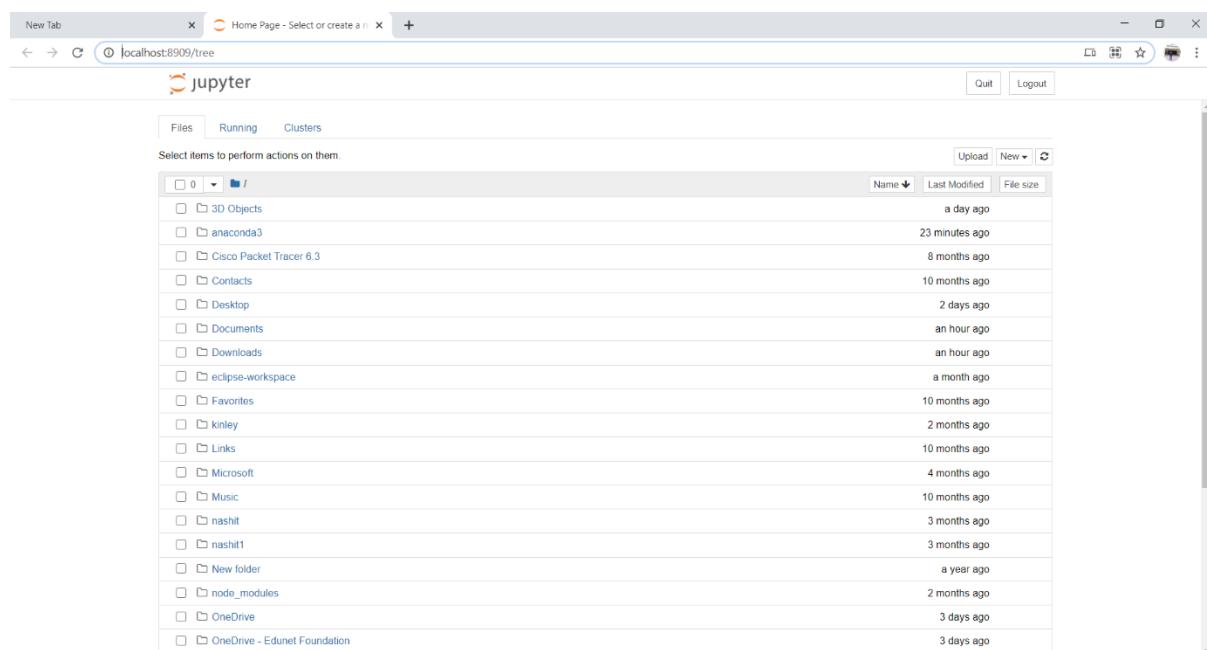
Since we already have Jupyter notebook installed on our systems, we can launch Jupyter notebook through Anaconda Navigator;

## 1. Open Anaconda Navigator



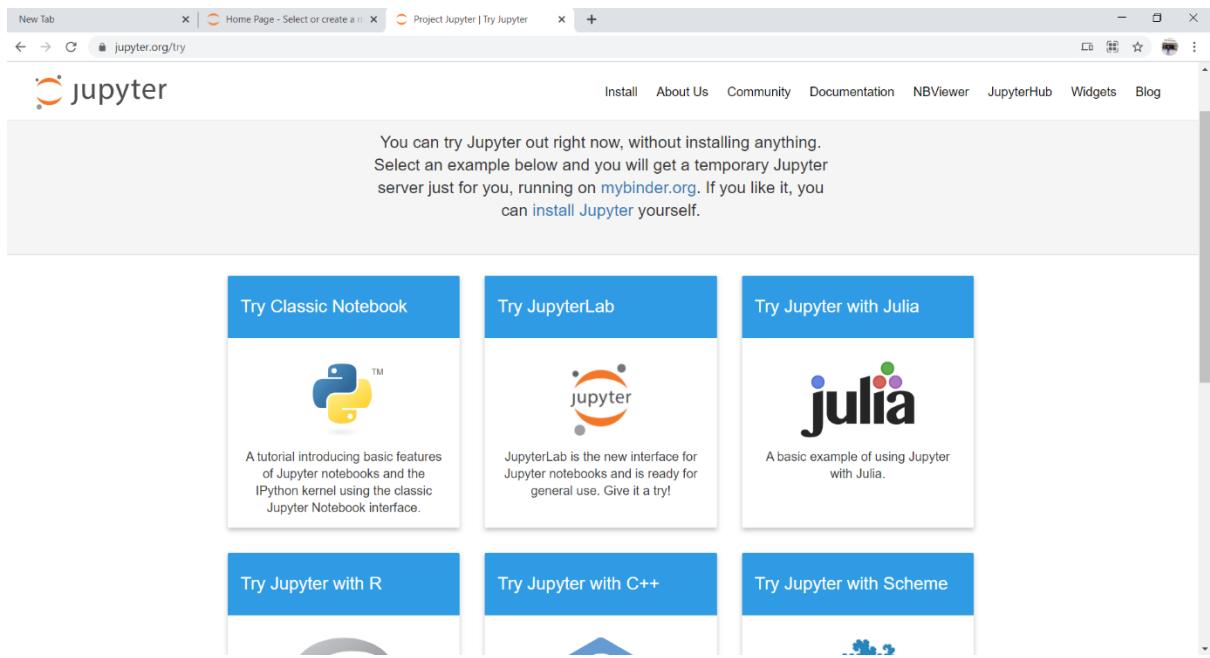
## 2. Launch Jupyter Notebook

3. This opens a local Jupyter Notebook on your machine, in the browser.



4. Alternatively, you may try online Jupyter notebook@

<https://jupyter.org/try>



You can try Jupyter out right now, without installing anything. Select an example below and you will get a temporary Jupyter server just for you, running on [mybinder.org](http://mybinder.org). If you like it, you can [install Jupyter yourself](#).

**Try Classic Notebook**  
 A tutorial introducing basic features of Jupyter notebooks and the IPython kernel using the classic Jupyter Notebook interface.

**Try JupyterLab**  
 JupyterLab is the new interface for Jupyter notebooks and is ready for general use. Give it a try!

**Try Jupyter with Julia**  
 A basic example of using Jupyter with Julia.

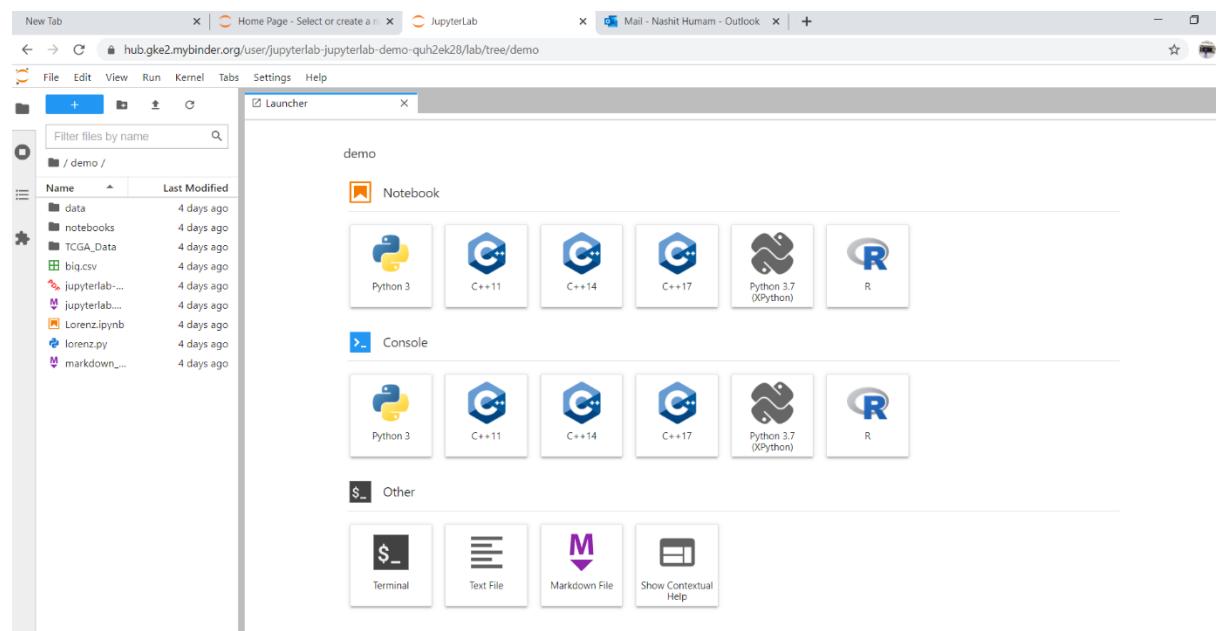
**Try Jupyter with R**  


**Try Jupyter with C++**  


**Try Jupyter with Scheme**  

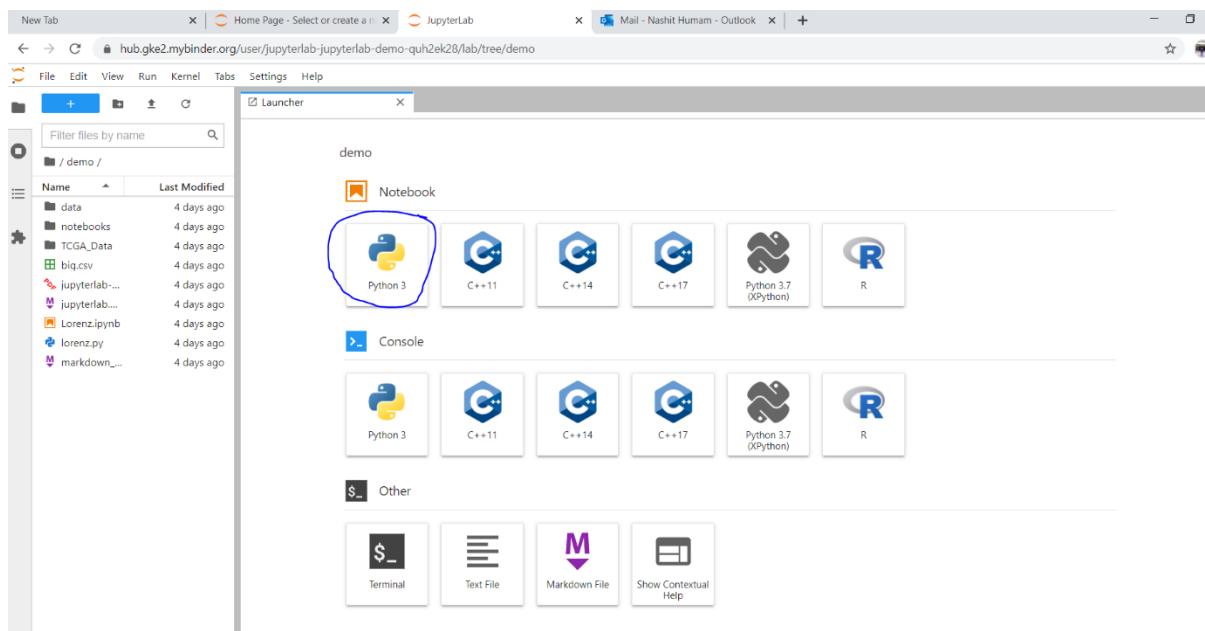

## 5. Try classic notebook

## 6. Choose the type of notebook you want to use;

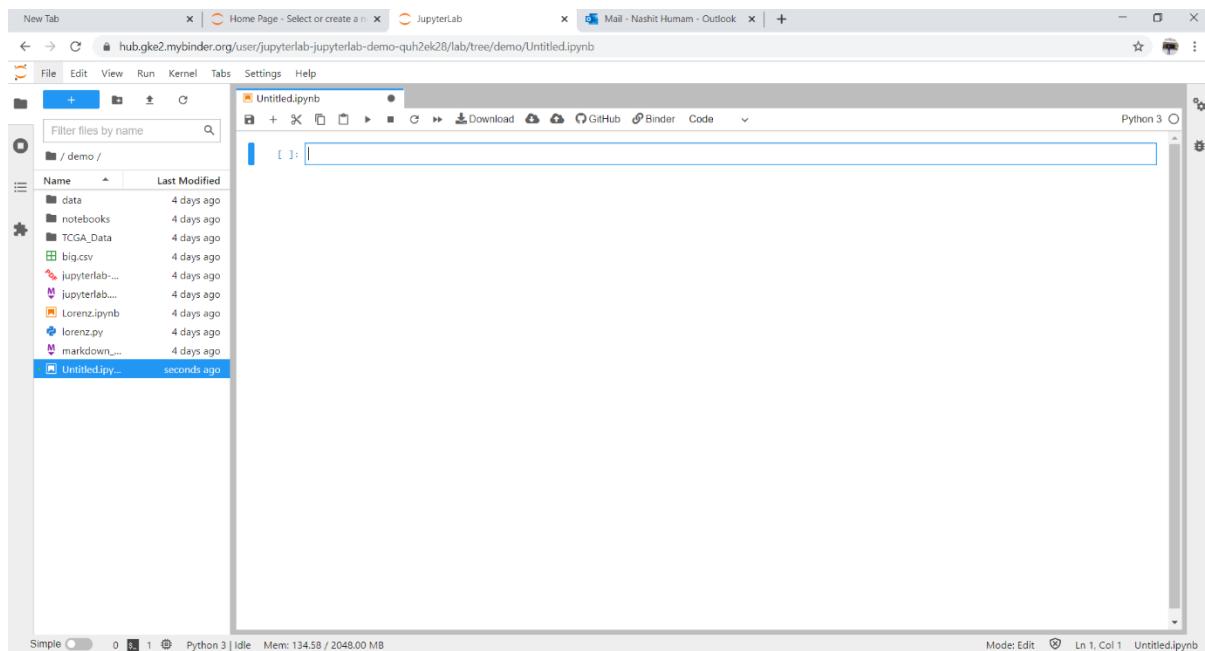


The screenshot shows the JupyterLab interface. On the left, there is a file tree with a list of files and directories under the path /demo/. The files include data, notebooks, TCGA\_Data, biq.csv, jupyterlab..., Lorenz.ipynb, Lorenz.py, and markdown.... The launcher on the right has sections for Notebook, Console, and Other. Under Notebook, there are icons for Python 3, C++11, C++14, C++17, Python 3.7 (XPython), and R. Under Console, there are icons for Python 3, C++11, C++14, C++17, Python 3.7 (XPython), and R. Under Other, there are icons for Terminal, Text File, Markdown File, and Show Contextual Help.

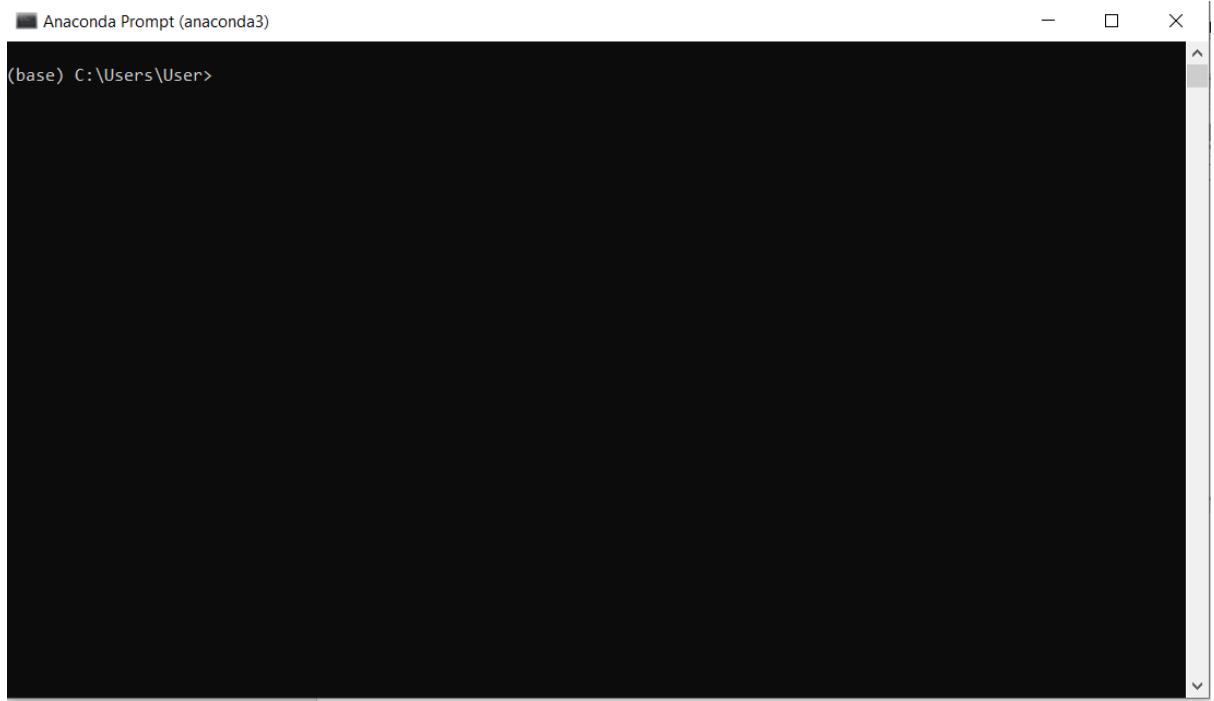
## 7. Try Notebook → Python



## 8. Try your codes;



## 9. Alternatively, you may install via console. Open Anaconda Prompt.



A screenshot of an Anaconda Prompt window titled "Anaconda Prompt (anaconda3)". The window is black and contains the text "(base) C:\Users\User>" at the top left. There are standard window controls (minimize, maximize, close) at the top right. A vertical scroll bar is visible on the right side of the window.

## 10. Use command

```
conda install -c conda-forge jupyterlab
```

```
■ Select Anaconda Prompt (anaconda3) - X ^

(base) C:\Users\User>conda install -c conda-forge jupyterlab
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\User\anaconda3

added / updated specs:
- jupyterlab

The following packages will be downloaded:

  package          |      build
  --              | -----
  conda-4.10.1    | py38haa244fe_0
  python_abi-3.8  |   1_cp38
  --              | Total:       3.1 MB

The following NEW packages will be INSTALLED:

  python_abi      conda-forge/win-64::python_abi-3.8-1_cp38

The following packages will be SUPERSEDED by a higher-priority channel:

  conda           pkgs/main::conda-4.10.1-py38haa95532_1 --> conda-forge::conda-4.10.1-py38haa244fe_0

Proceed ([y]/n)?


Downloading and Extracting Packages
conda-4.10.1      | 3.1 MB      | #####| 100%
python_abi-3.8    | 4 KB        | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\User>
```

## 11. Run Jupyter Lab

jupyter-lab

```
[I 16:10:38.130 LabApp] or http://127.0.0.1:8814/?token=6bcd3c5401dd24bedf57e84df890f519f21d8d6204a80bb
[I 16:10:38.130 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:10:38.301 LabApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/User/AppData/Roaming/jupyter/runtime/nbserver-6392-open.html
Or copy and paste one of these URLs:
    http://localhost:8814/?token=6bcd3c5401dd24bedf57e84df890f519f21d8d6204a80bb
    or http://127.0.0.1:8814/?token=6bcd3c5401dd24bedf57e84df890f519f21d8d6204a80bb
[I 16:10:51.924 LabApp] Build is up to date
[I 16:11:38.310 LabApp] Interrupted...
[I 16:11:38.310 LabApp] Shutting down 0 kernels
[I 16:11:38.310 LabApp] Shutting down 0 terminals

(base) C:\Users\User>
(base) C:\Users\User>
(base) C:\Users\User>
(base) C:\Users\User>
(base) C:\Users\User>
(base) C:\Users\User>
(base) C:\Users\User>jupyter-lab
[I 16:11:51.486 LabApp] The port 8888 is already in use, trying another port.
[I 16:11:51.486 LabApp] The port 8889 is already in use, trying another port.
[I 16:11:51.486 LabApp] The port 8890 is already in use, trying another port.
[I 16:11:51.486 LabApp] The port 8891 is already in use, trying another port.
[I 16:11:51.486 LabApp] The port 8892 is already in use, trying another port.
[I 16:11:52.077 LabApp] JupyterLab extension loaded from C:\Users\User\anaconda3\lib\site-packages\jupyterlab
[I 16:11:52.077 LabApp] JupyterLab application directory is C:\Users\User\anaconda3\share\jupyter\lab
[I 16:11:52.077 LabApp] Serving notebooks from local directory: C:\Users\User
[I 16:11:52.077 LabApp] Jupyter Notebook 6.1.4 is running at:
[I 16:11:52.077 LabApp] http://localhost:8981/?token=0d249e2dc55ed3222d8f291b192e9de48d63e172166df0e4
[I 16:11:52.093 LabApp] or http://127.0.0.1:8981/?token=0d249e2dc55ed3222d8f291b192e9de48d63e172166df0e4
[I 16:11:52.093 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:11:52.247 LabApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/User/AppData/Roaming/jupyter/runtime/nbserver-4236-open.html
Or copy and paste one of these URLs:
    http://localhost:8981/?token=0d249e2dc55ed3222d8f291b192e9de48d63e172166df0e4
    or http://127.0.0.1:8981/?token=0d249e2dc55ed3222d8f291b192e9de48d63e172166df0e4
[I 16:11:57.703 LabApp] Build is up to date
```

12. This opens the Notebook in the browser.

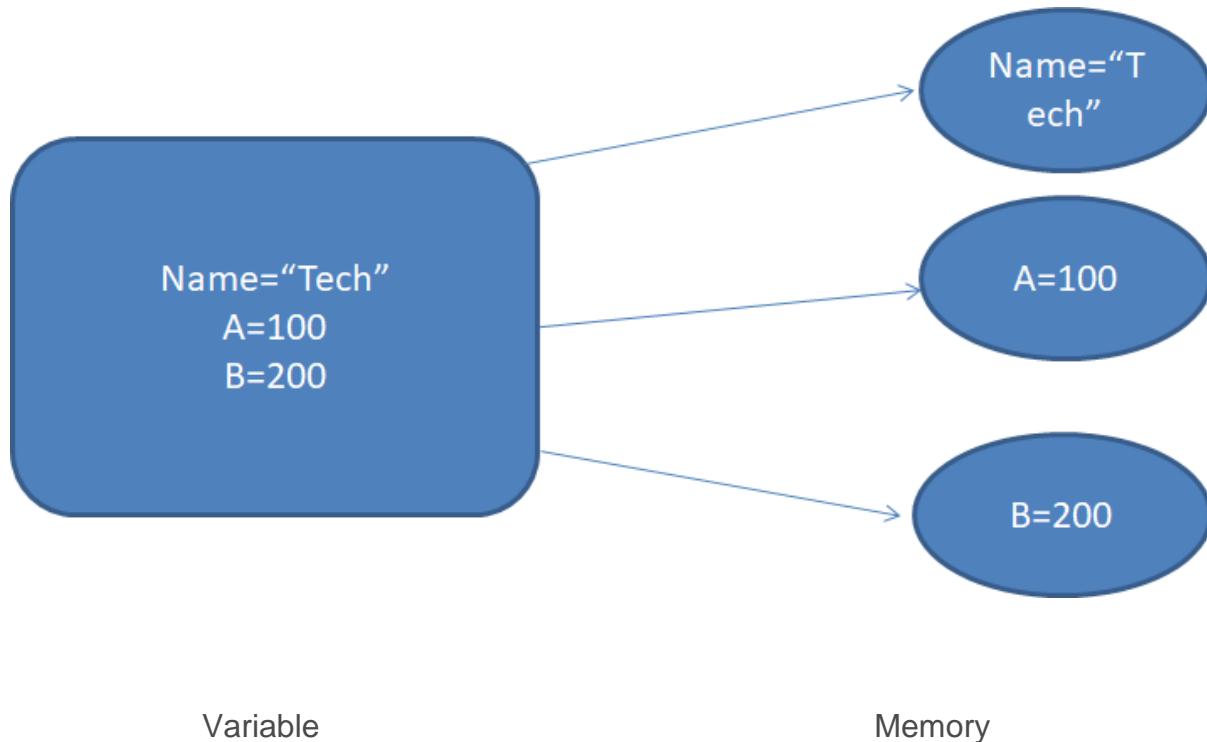
So, after creating our environment for programming let's start with some basic building blocks of Python language.

## 2.10 Variables in Python

Variables are containers for storing data values.

### ***What Are Variables in Python?***

Variables and data types in python as the name suggests are the values that vary. In a programming language, a variable is a memory location where you store a value. The value that you have stored may change in the future according to the specifications.



Variable in python is created as soon as a value is assigned to it. It does not need any additional commands to declare a variable in python.

After introducing to python lets install anaconda on Linux and windows operating systems to perform some data analysis and machine learning stuff.

## 2.11 Advanced Datatypes in Python

Data type defines the format, sets the upper & lower bounds of the data so that a program could use it appropriately.

In Python, we don't need to declare a variable with explicitly mentioning the data type. This feature is famously known as dynamic typing.

Python determines the type of a literal directly from the syntax at runtime. For example – the quotes mark the declaration of a string value, square brackets represent a list and curly brackets for a dictionary. Also, the non-decimal numbers will get assigned to Integer type whereas the ones with a decimal point will be a float.

Below is the list of important data types that are commonly used in Python.

- Booleans
- Numbers

- Strings
- Bytes
- Lists
- Tuples
- Sets
- Dictionaries

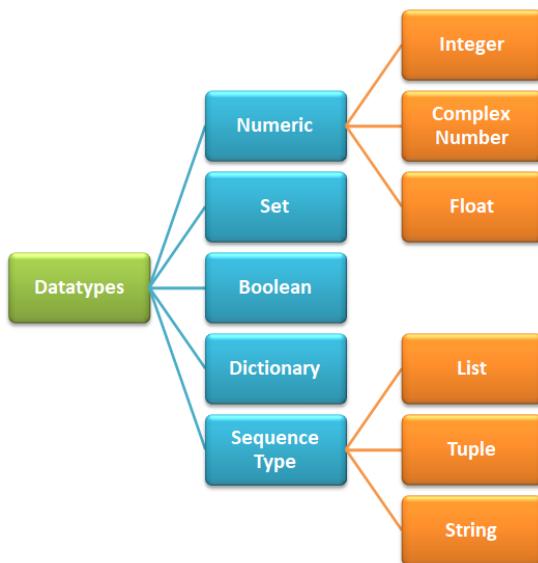


Fig: Datatypes in Python

In Python, numeric data type represents the data which has numeric value. Numeric value can be integer, floating number or even complex numbers. These values are defined as int, float and complex class in Python.

- **Integers** – This value is represented by int class. It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be.
- **Float** – This value is represented by float class. It is a real number with floating point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation.
- **Complex Numbers** – Complex number is represented by complex class. It is specified as (real part) + (imaginary part)j. For example – 2+3j

```
a = 5
print("Type of a: ", type(a))

b = 5.0
print("\nType of b: ", type(b))

c = 2 + 4j
print("\nType of c: ", type(c))
```

Type of a: <class 'int'>

Type of b: <class 'float'>

Type of c: <class 'complex'>

## Sequence Type

In Python, sequence is the ordered collection of similar or different data types. Sequences allows to store multiple values in an organized and efficient fashion. There are several sequence types in Python –

- String
- List
- Tuple

### 1) String

In Python, Strings are arrays of bytes representing Unicode characters. A string is a collection of one or more characters put in a single quote, double-quote or triple quote. In python there is no character data type, a character is a string of length one. It is represented by “str” class.

#### Creating String

Strings in Python can be created using single quotes or double quotes or even triple quotes.

```
[4]: String1 = 'Welcome to Python World'
      print("String with the use of Single Quotes: ")
      print(String1)

      # Creating a String
      # with double Quotes
      String1 = "Welcome to Python's World"
      print("\nString with the use of Double Quotes: ")
      print(String1)
      print(type(String1))

      # Creating a String
      # with triple Quotes
      String1 = '''Welcome to the World of "Python"'''
      print("\nString with the use of Triple Quotes: ")
      print(String1)
      print(type(String1))

      # Creating String with triple
      # Quotes allows multiple lines
      String1 = '''Welcome
                  To
                  Python'''
      print("\nCreating a multiline String: ")
      print(String1)
```

String with the use of Single Quotes:  
Welcome to Python World

String with the use of Double Quotes:  
Welcome to Python's World  
<class 'str'>

String with the use of Triple Quotes:  
Welcome to the World of "Python"  
<class 'str'>

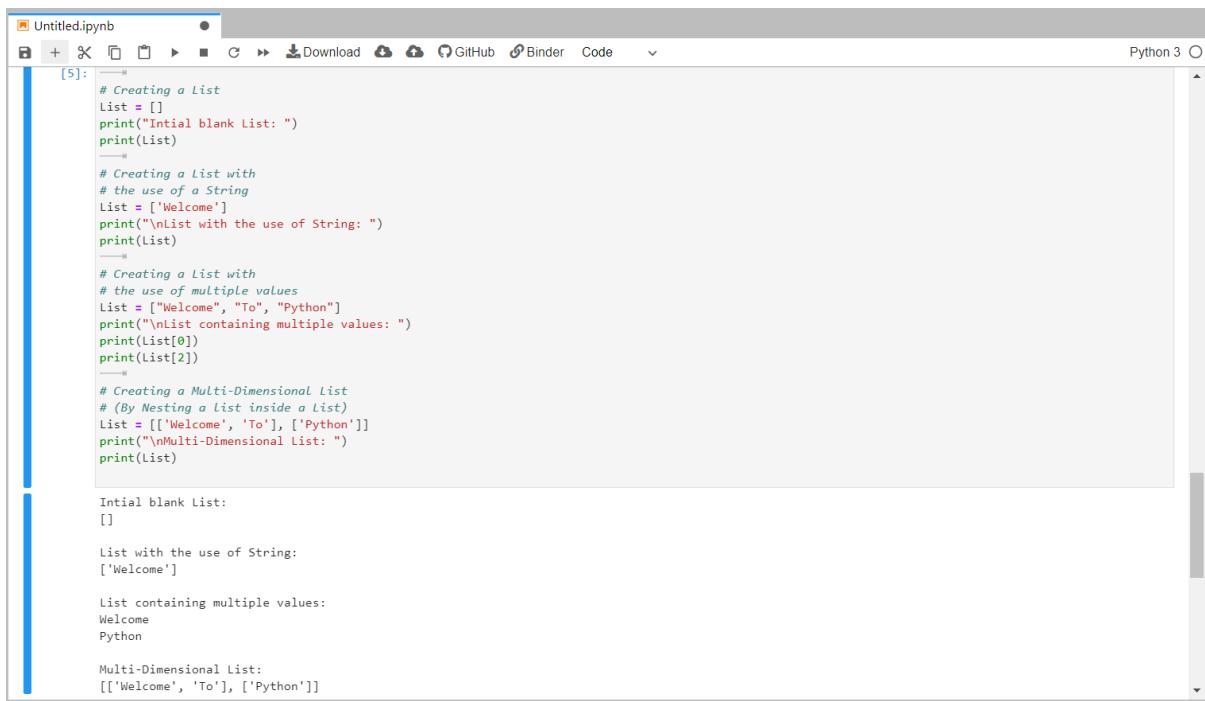
Creating a multiline String:

## 2) List

Lists are just like the arrays, declared in other languages which is a ordered collection of data. It is very flexible as the items in a list do not need to be of the same type.

### Creating List

Lists in Python can be created by just placing the sequence inside the square brackets [ ].



The screenshot shows a Jupyter Notebook cell titled "Untitled.ipynb" with the code cell number [5]. The code demonstrates four ways to create lists:

```
# Creating a List
List = []
print("Initial blank List: ")
print(List)

# Creating a List with
# the use of a String
List = ['Welcome']
print("\nList with the use of String: ")
print(List)

# Creating a List with
# the use of multiple values
List = ["Welcome", "To", "Python"]
print("\nList containing multiple values: ")
print(List[0])
print(List[2])

# Creating a Multi-Dimensional List
# (By Nesting a list inside a List)
List = [[['Welcome', 'To'], ['Python']]]
print("\nMulti-Dimensional List: ")
print(List)
```

The output of the code is:

```
Initial blank List:
[]

List with the use of String:
['Welcome']

List containing multiple values:
Welcome
Python

Multi-Dimensional List:
[[['Welcome', 'To'], ['Python']]]
```

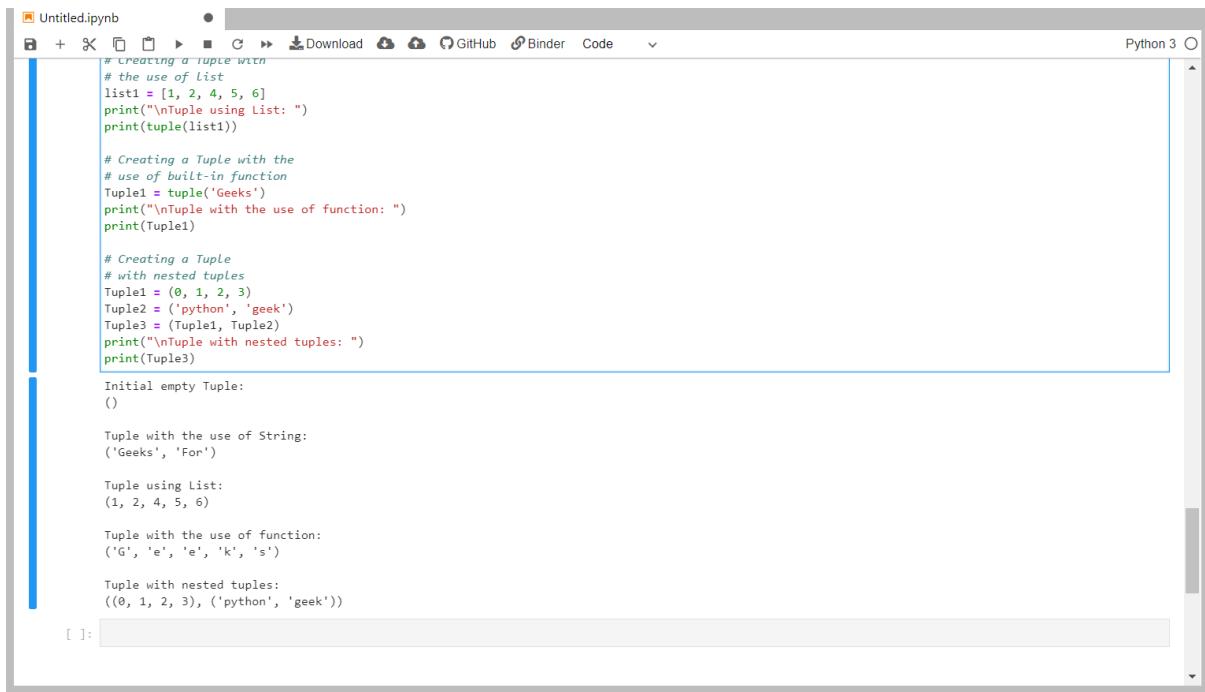
### 3) Tuple

Just like list, tuple is also an ordered collection of Python objects. The only difference between tuple and list is that tuples are immutable i.e. tuples cannot be modified after it is created. It is represented by `tuple` class.

#### Creating Tuple

In Python, tuples are created by placing a sequence of values separated by 'comma' with or without the use of parentheses for grouping of the data sequence. Tuples can contain any number of elements and of any datatype (like strings, integers, list, etc.).

**Note:** Tuples can also be created with a single element, but it is a bit tricky. Having one element in the parentheses is not sufficient, there must be a trailing 'comma' to make it a tuple.



```
# Creating a tuple with
# the use of List
list1 = [1, 2, 4, 5, 6]
print("\nTuple using List: ")
print(tuple(list1))

# Creating a Tuple with the
# use of built-in function
tuple1 = tuple('Geeks')
print("\nTuple with the use of function: ")
print(tuple1)

# Creating a Tuple
# with nested tuples
tuple1 = (0, 1, 2, 3)
tuple2 = ('python', 'geek')
tuple3 = (tuple1, tuple2)
print("\nTuple with nested tuples: ")
print(tuple3)

Initial empty Tuple:
()

Tuple with the use of String:
('Geeks', 'For')

Tuple using List:
(1, 2, 4, 5, 6)

Tuple with the use of function:
('G', 'e', 'e', 'k', 's')

Tuple with nested tuples:
((0, 1, 2, 3), ('python', 'geek'))
```

#### 4) Boolean

Data type with one of the two built-in values, `True` or `False`. Boolean objects that are equal to `True` are truthy (true), and those equal to `False` are falsy (false). But non-Boolean objects can be evaluated in Boolean context as well and determined to be true or false. It is denoted by the class `bool`.

**Note** – `True` and `False` with capital ‘T’ and ‘F’ are valid booleans otherwise python will throw an error.

```
[10]:  
  
# Python program to  
# demonstrate boolean  
  
typex = bool(5)  
  
#display x:  
print(x)  
  
#display the data type of x:  
print(type(x))
```

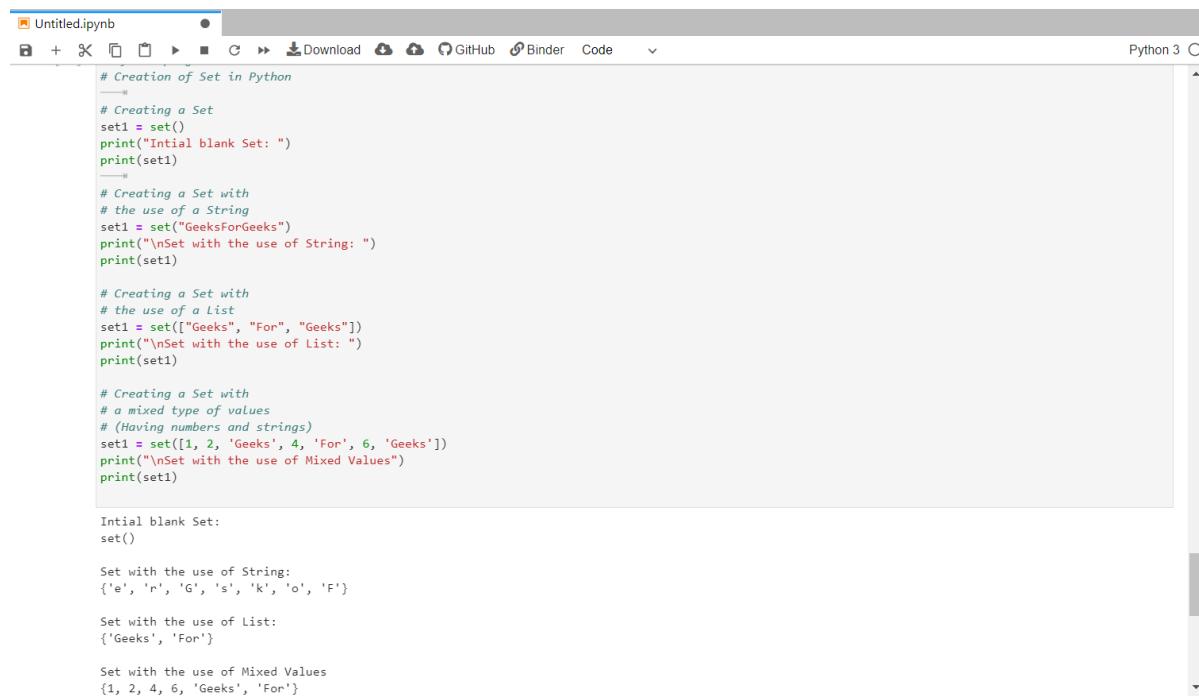
```
True  
<class 'bool'>
```

## 5) Set

In Python, Set is an unordered collection of data type that is iterable, mutable and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

### Creating Sets

Sets can be created by using the built-in `set()` function with an iterable object or a sequence by placing the sequence inside curly braces, separated by ‘comma’. Type of elements in a set need not be the same, various mixed-up data type values can also be passed to the set.



```
# Creation of Set in Python
# Creating a Set
set1 = set()
print("Initial blank Set: ")
print(set1)

# Creating a Set with
# the use of a String
set1 = set("GeeksForGeeks")
print("\nSet with the use of String: ")
print(set1)

# Creating a Set with
# the use of a list
set1 = set(["Geeks", "For", "Geeks"])
print("\nSet with the use of List: ")
print(set1)

# Creating a Set with
# a mixed type of values
# (Having numbers and strings)
set1 = set([1, 2, 'Geeks', 4, 'For', 6, 'Geeks'])
print("\nSet with the use of Mixed Values")
print(set1)
```

Initial blank Set:  
set()

Set with the use of String:  
{'e', 'n', 'G', 's', 'k', 'o', 'F'}

Set with the use of List:  
{'Geeks', 'For'}

Set with the use of Mixed Values  
{1, 2, 4, 6, 'Geeks', 'For'}

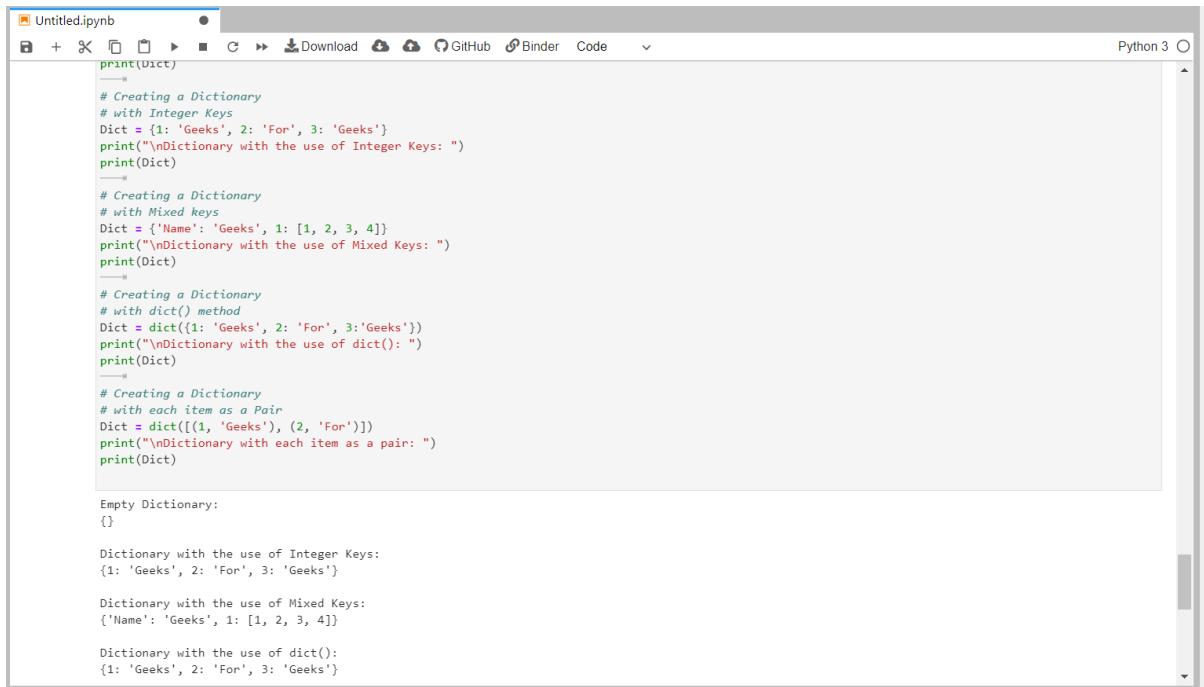
## 6) Dictionary

Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds `key:value` pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a ‘comma’.

### Creating Dictionary

In Python, a Dictionary can be created by placing a sequence of elements within curly {} braces, separated by ‘comma’. Values in a dictionary can be of any datatype and can be duplicated, whereas keys can't be repeated and must be immutable. Dictionary can also be created by the built-in function `dict()`. An empty dictionary can be created by just placing it to curly braces{}.

**Note** – Dictionary keys are case sensitive, same name but different cases of Key will be treated distinctly.



```
print(Dict)
# Creating a Dictionary
# with Integer Keys
Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}
print("\nDictionary with the use of Integer Keys: ")
print(Dict)

# Creating a Dictionary
# with Mixed keys
Dict = {'Name': 'Geeks', 1: [1, 2, 3, 4]}
print("\nDictionary with the use of Mixed Keys: ")
print(Dict)

# Creating a Dictionary
# with dict() method
Dict = dict({1: 'Geeks', 2: 'For', 3:'Geeks'})
print("\nDictionary with the use of dict(): ")
print(Dict)

# Creating a Dictionary
# with each item as a Pair
Dict = dict([(1, 'Geeks'), (2, 'For')])
print("\nDictionary with each item as a pair: ")
print(Dict)

Empty Dictionary:
{}

Dictionary with the use of Integer Keys:
{1: 'Geeks', 2: 'For', 3: 'Geeks'}

Dictionary with the use of Mixed Keys:
{'Name': 'Geeks', 1: [1, 2, 3, 4]}

Dictionary with the use of dict():
{1: 'Geeks', 2: 'For', 3: 'Geeks'}
```

## 2.12 Functions & Methods in Python

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

Python gives us many built-in functions like `print()`, etc. but you can also create your own functions. These functions are called *user-defined functions*.

### Defining a Function

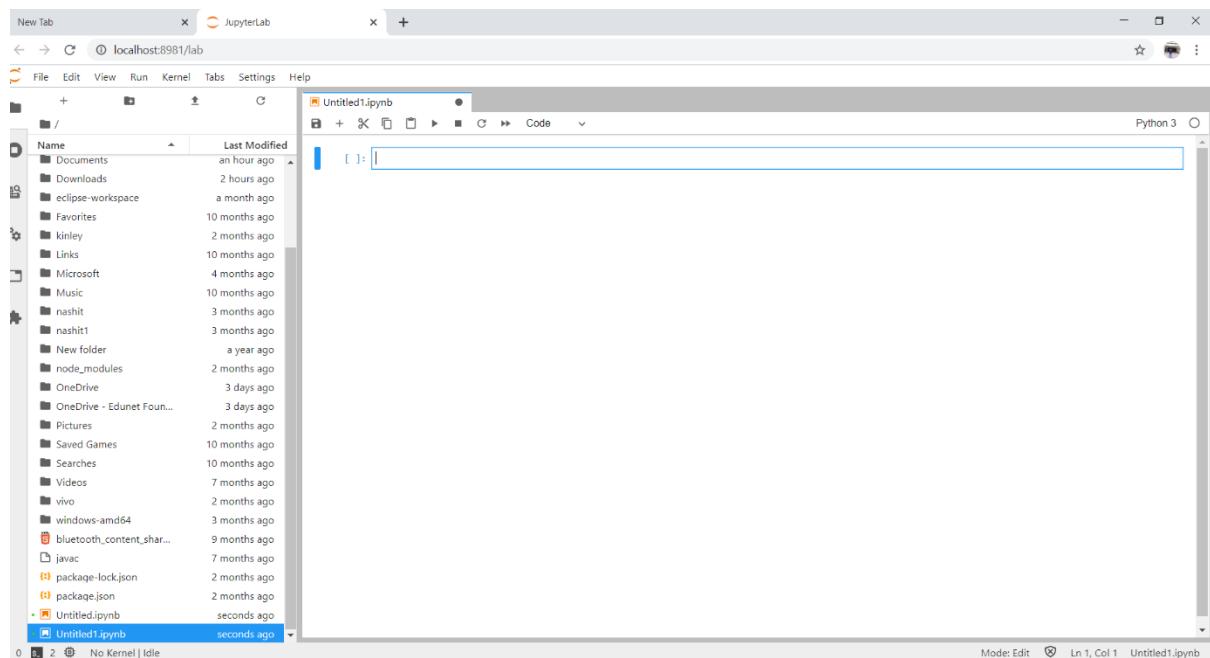
You can define functions to provide the required functionality. Here are simple rules to define a function in Python.

- Function blocks begin with the keyword **def** followed by the function name and parentheses ( ).
- Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
- The first statement of a function can be an optional statement - the documentation string of the function or *docstring*.
- The code block within every function starts with a colon (:) and is indented.

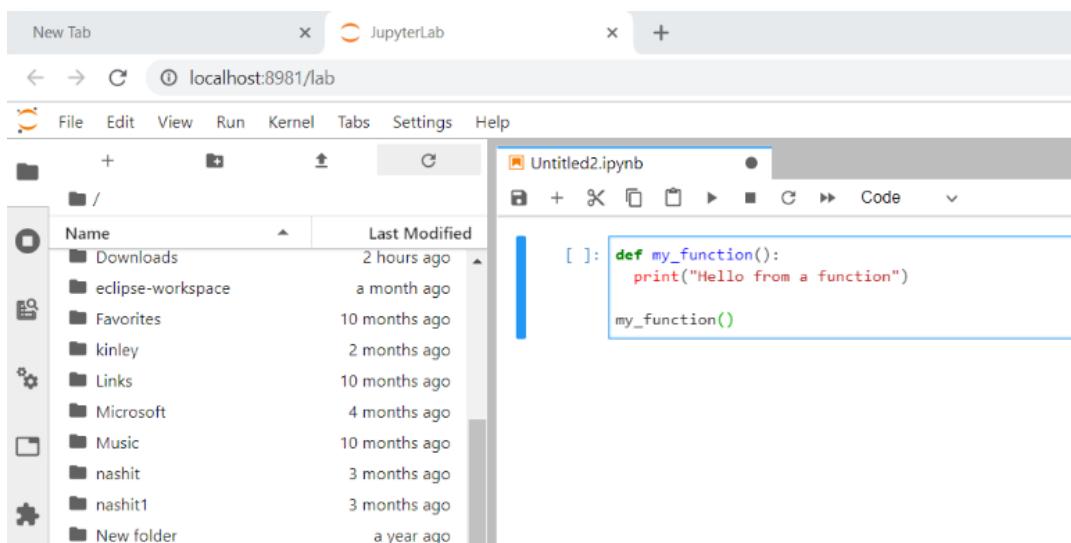
- The statement `return [expression]` exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as `return None`.

An example can be;

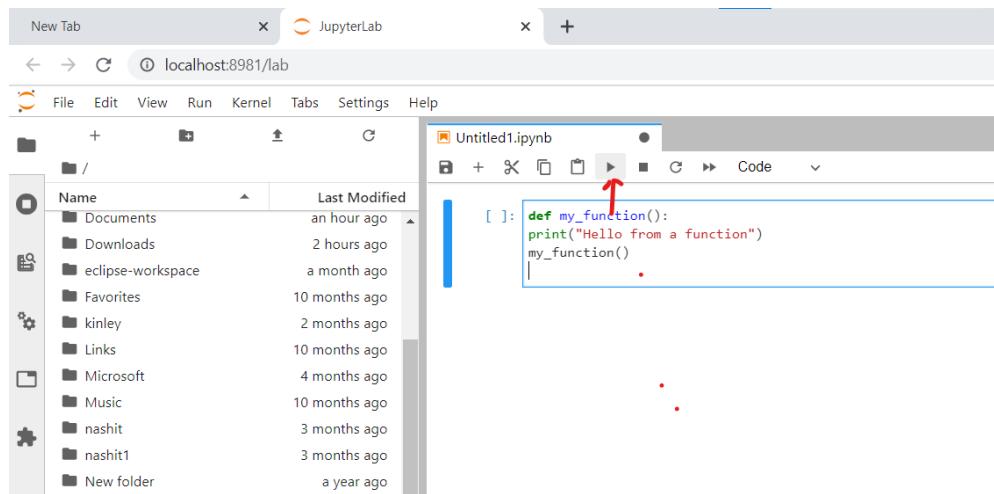
### 1. Open a blank notebook.



### 2. Try the following code;



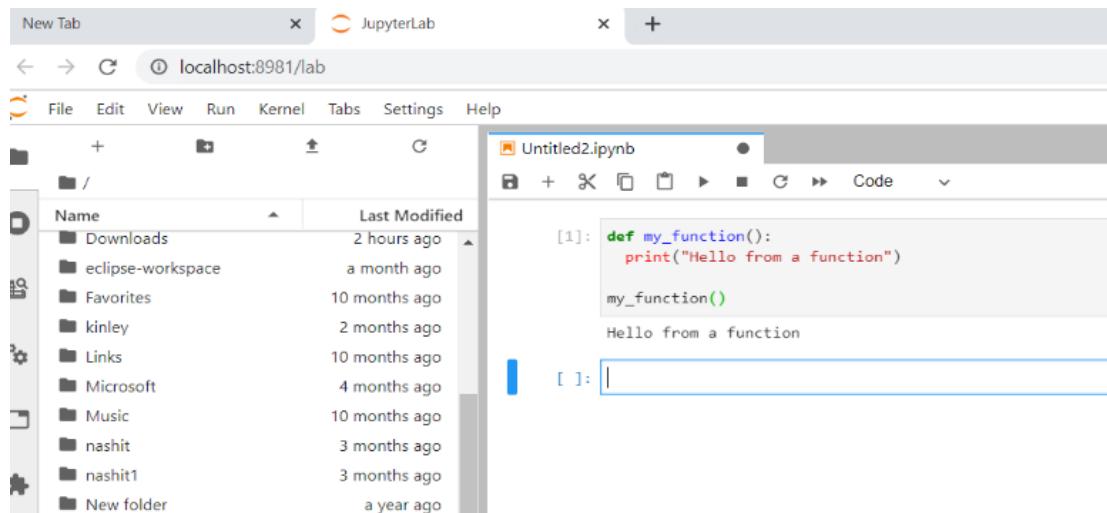
### 3. Run the code, using Run button;



A screenshot of the JupyterLab interface. On the left is a file browser showing a list of files and folders. On the right is a notebook editor with an open file named 'Untitled1.ipynb'. A code cell contains the following Python code:

```
[ ]: def my_function():
    print("Hello from a function")
my_function()
```

#### 4. The output;



A screenshot of the JupyterLab interface. On the left is a file browser showing a list of files and folders. On the right is a notebook editor with an open file named 'Untitled2.ipynb'. A code cell contains the following Python code, which has been executed:

```
[1]: def my_function():
    print("Hello from a function")
my_function()
```

The output of the code is displayed below the cell:

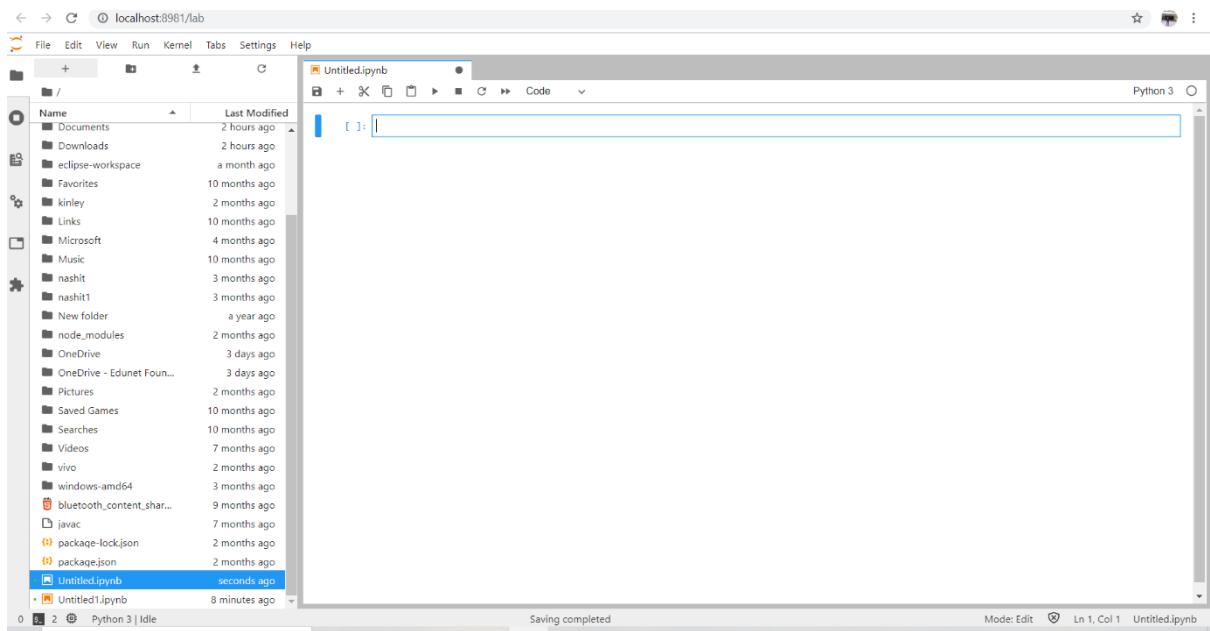
```
Hello from a function
```

## Method

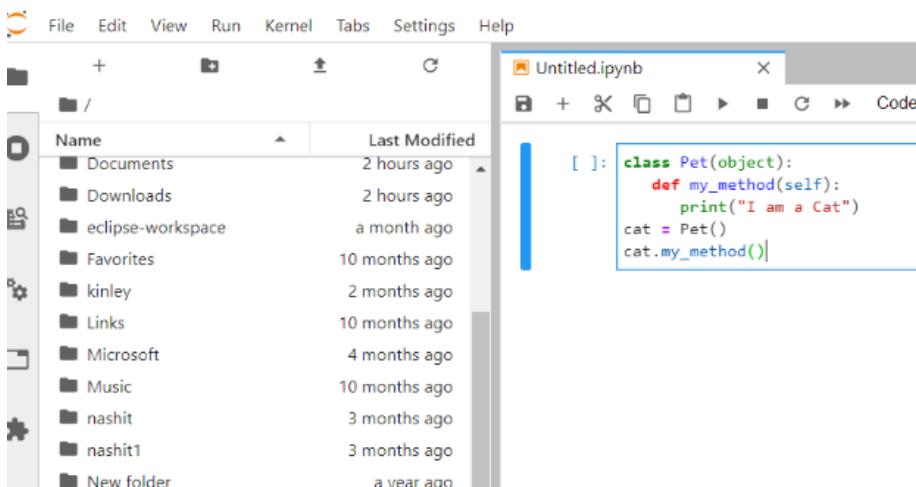
A method in python is somewhat similar to a function, except it is associated with object/classes. Methods in python are very similar to functions except for two major differences.

- The method is implicitly used for an object for which it is called.
- The method is accessible to data that is contained within the class.

1. Open a new blank notebook.

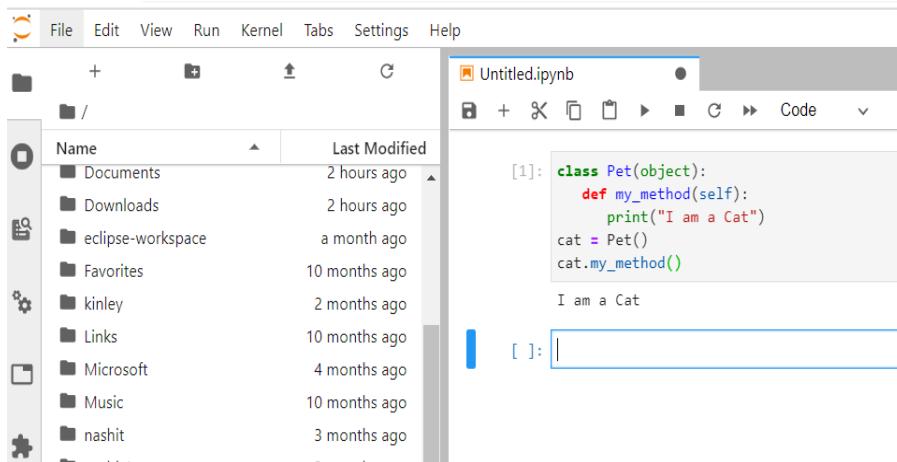


## 2. Write a piece of code using method()



## 3. Run locally

## 4. Output



The screenshot shows a Jupyter Notebook interface. On the left is a file browser with a list of files and folders. In the center, a code cell contains Python code defining a class `Pet` with a method `my_method`. The output of the cell shows the string "I am a Cat".

Name	Last Modified
Documents	2 hours ago
Downloads	2 hours ago
eclipse-workspace	a month ago
Favorites	10 months ago
kinley	2 months ago
Links	10 months ago
Microsoft	4 months ago
Music	10 months ago
nashit	3 months ago

```
[1]: class Pet(object):
    def my_method(self):
        print("I am a Cat")
cat = Pet()
cat.my_method()
```

I am a Cat

Now, after understanding the concept of function in python let's get familiar with conditional operators and looping statements in Python.

## 2.13 Condition & Loop in Python

### Condition

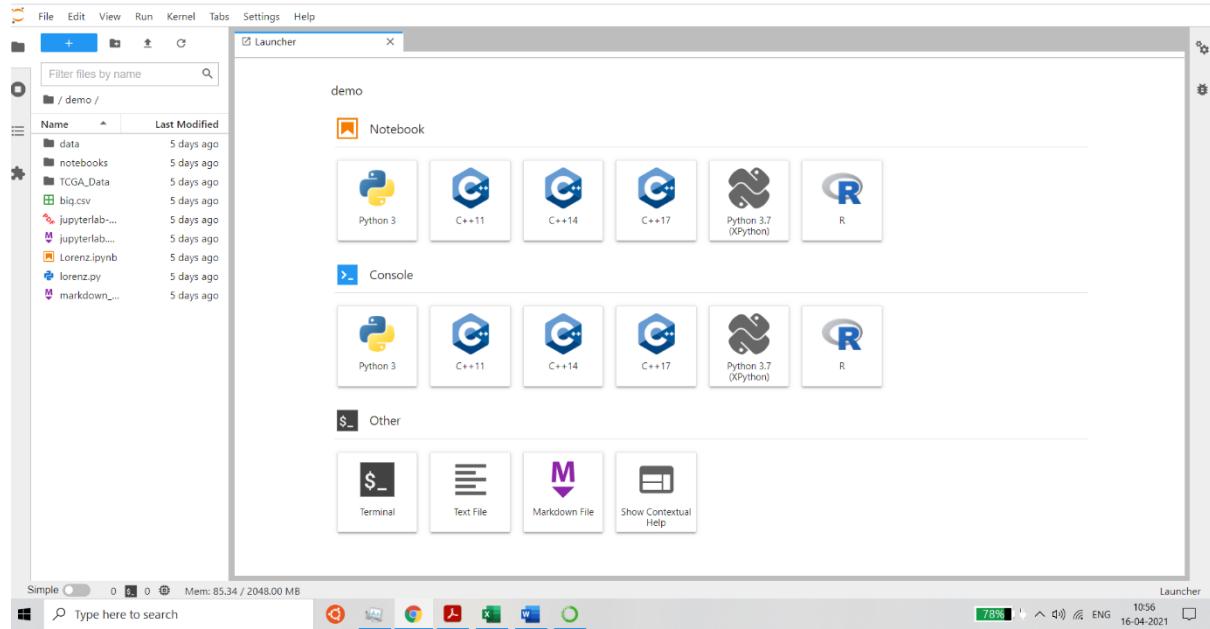
Python supports the usual logical conditions from mathematics:

- Equals: `a == b`
- Not Equals: `a != b`
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`

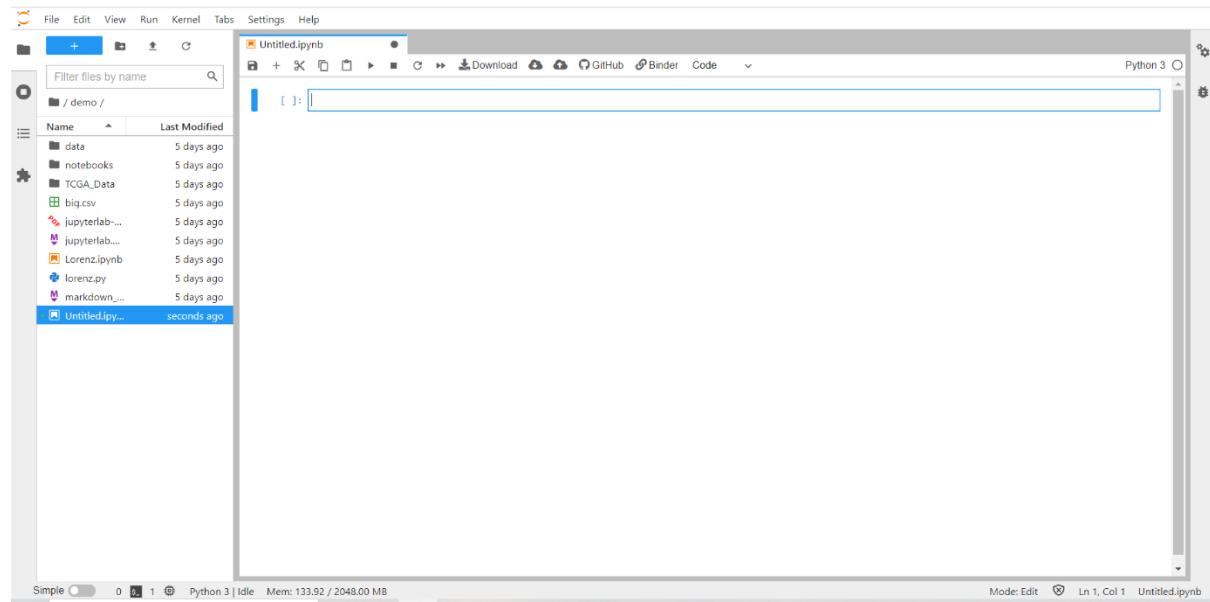
These conditions can be used in several ways, most commonly in "if statements" and loops.

An "if statement" is written by using the `if` keyword.

1. Open Jupyter Notebook

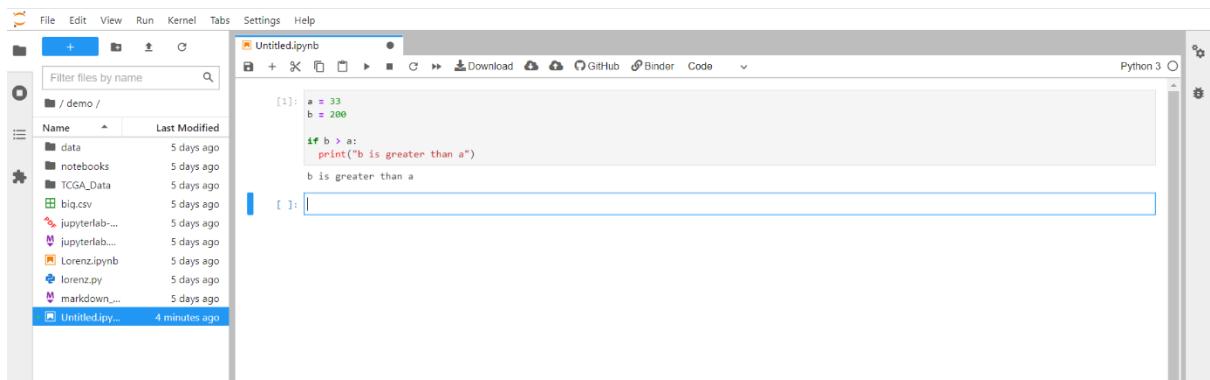


## 2. Go for Python



## 3. Try the code for Condition.

(Take care of the indentation (whitespace at the beginning of a line) to define scope in the code)

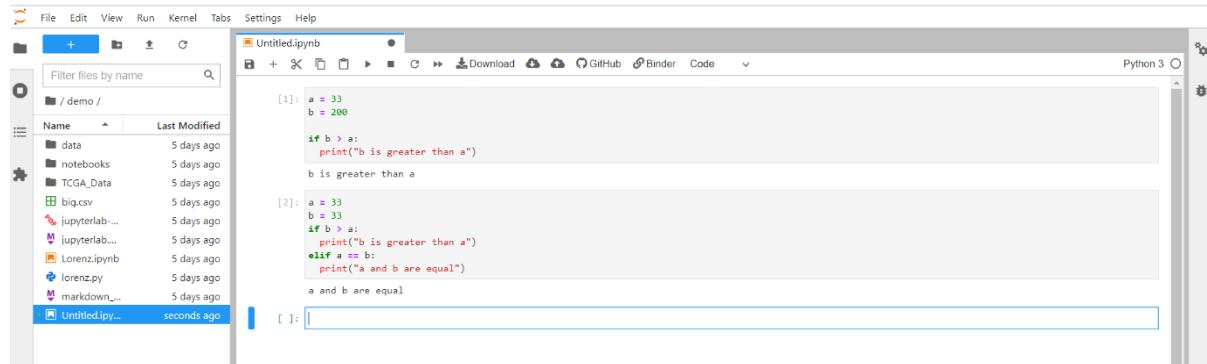


```
[1]: a = 33
b = 200

if b > a:
    print("b is greater than a")
b is greater than a
```

## *Elif*

The **elif** keyword is python's way of saying "if the previous conditions were not true, then try this condition".



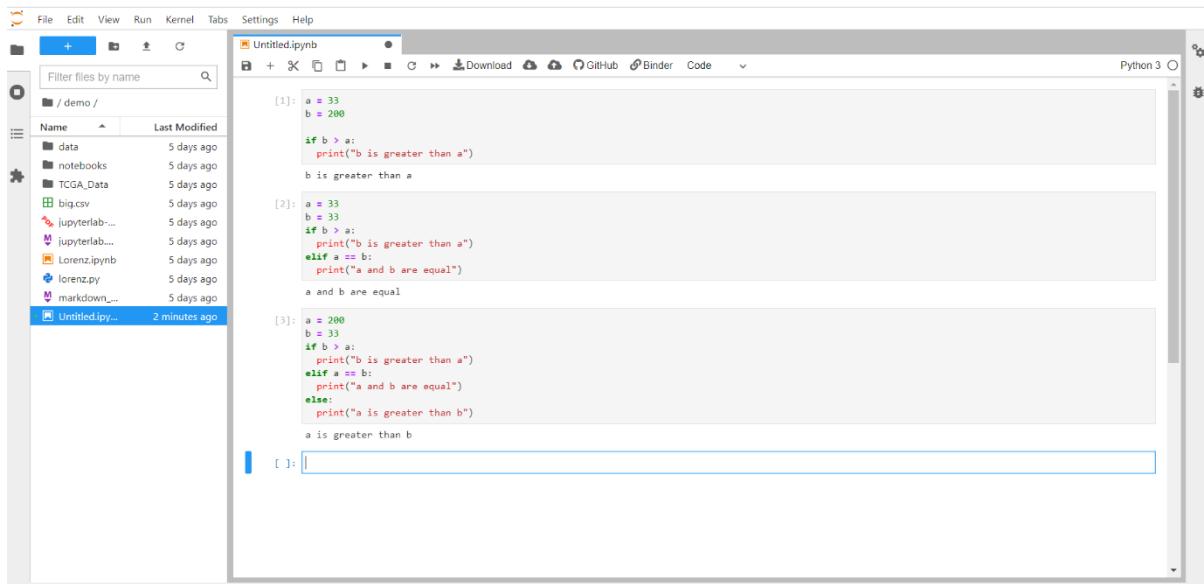
```
[1]: a = 33
b = 200

if b > a:
    print("b is greater than a")
b is greater than a

[2]: a = 33
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
a and b are equal
```

## *Else*

The **else** keyword catches anything which isn't caught by the preceding conditions.



The screenshot shows a Jupyter Notebook interface with three code cells:

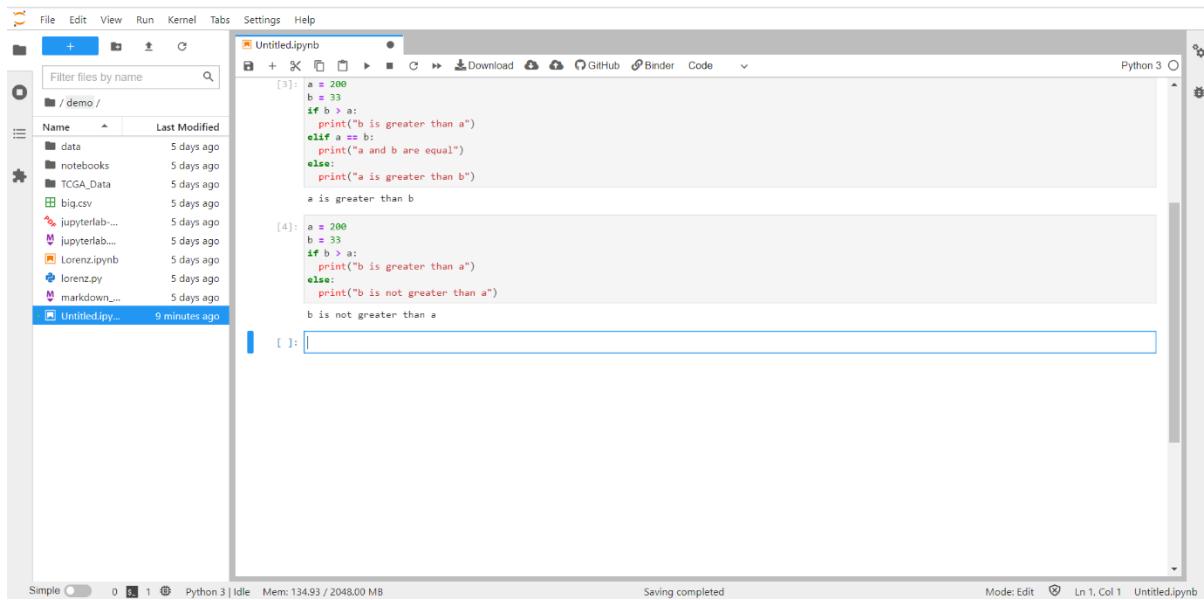
```
[1]: a = 33
b = 200

if b > a:
    print("b is greater than a")
b is greater than a

[2]: a = 33
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
a and b are equal

[3]: a = 200
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
else:
    print("a is greater than b")
a is greater than b
```

### *else without the elif*



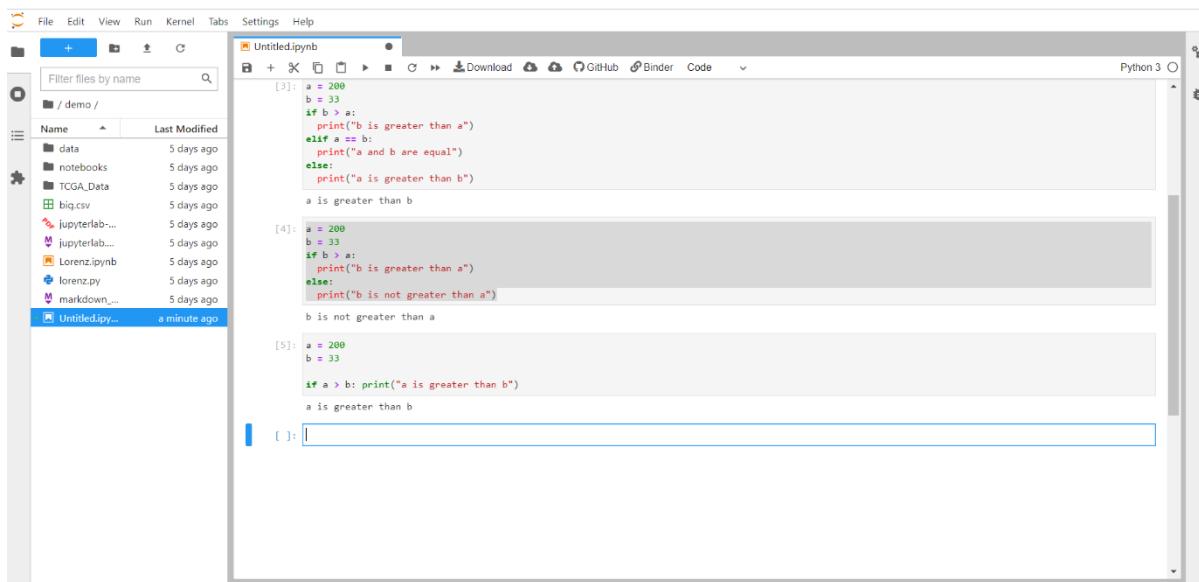
The screenshot shows a Jupyter Notebook interface with two code cells:

```
[3]: a = 200
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
else:
    print("a is greater than b")
a is greater than b

[4]: a = 200
b = 33
if b > a:
    print("b is greater than a")
else:
    print("b is not greater than a")
b is not greater than a
```

### *If*

If you have only one statement to execute, you can put it on the same line as the if statement.



The screenshot shows a Jupyter Notebook interface with a sidebar containing files like 'Untitled.ipynb', 'data', 'notebooks', 'TCGA\_Data', 'biq.csv', 'jupyterlab...', 'jupyterlab...', 'Lorenz.ipynb', 'lorenz.py', and 'markdown\_...'. The main area displays Python code in cells:

```

[3]: a = 200
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
else:
    print("a is greater than b")
a is greater than b

[4]: a = 200
b = 33
if b > a:
    print("b is greater than a")
else:
    print("b is not greater than a")
b is not greater than a

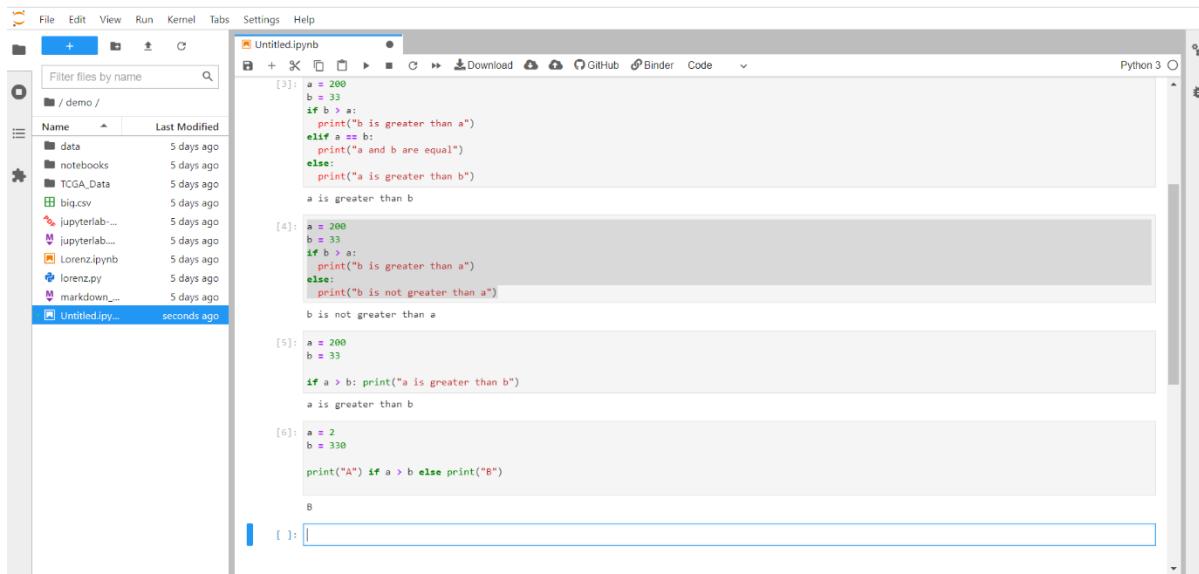
[5]: a = 200
b = 33

if a > b: print("a is greater than b")
a is greater than b

```

## If ... Else

If you have only one statement to execute, one for if, and one for else, you can put it all on the same line:



The screenshot shows a Jupyter Notebook interface with a sidebar containing files like 'Untitled.ipynb', 'data', 'notebooks', 'TCGA\_Data', 'biq.csv', 'jupyterlab...', 'jupyterlab...', 'Lorenz.ipynb', 'lorenz.py', and 'markdown\_...'. The main area displays Python code in cells:

```

[3]: a = 200
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
else:
    print("a is greater than b")
a is greater than b

[4]: a = 200
b = 33
if b > a:
    print("b is greater than a")
else:
    print("b is not greater than a")
b is not greater than a

[5]: a = 200
b = 33

if a > b: print("a is greater than b")
a is greater than b

[6]: a = 2
b = 330

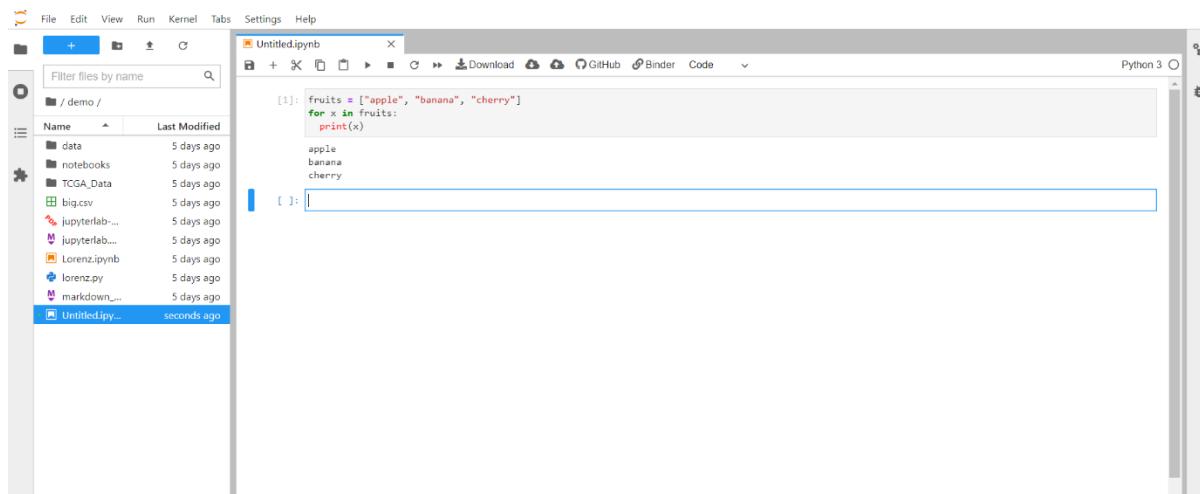
print("A") if a > b else print("B")
B

```

## Loops

A **for** loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string). This is less like the **for** keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.

With the **for** loop we can execute a set of statements, once for each item in a list, tuple, set etc.

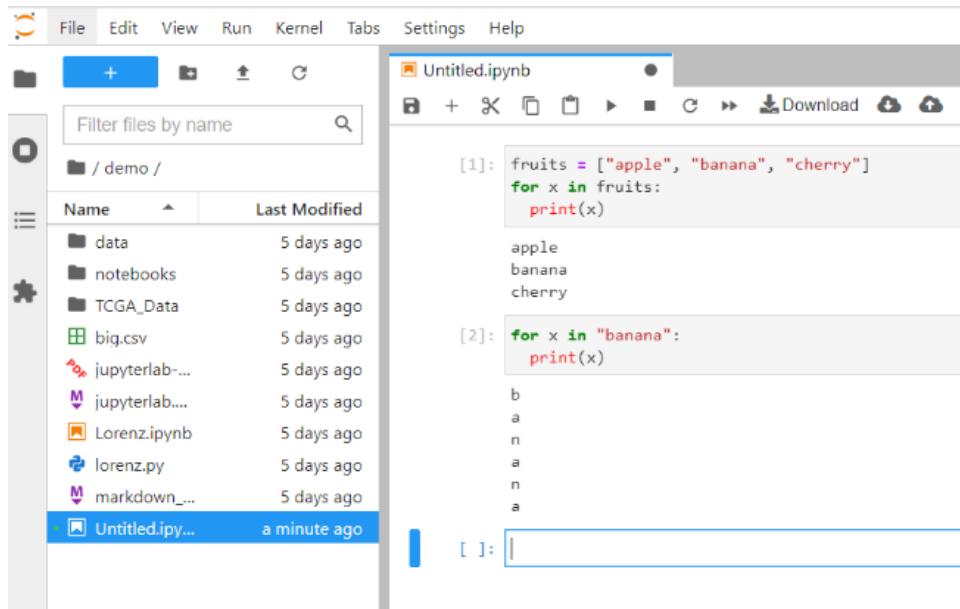


```
[1]: fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)

apple
banana
cherry
```

### *Looping Through a String*

Even strings are iterable objects, they contain a sequence of characters:



```
[1]: fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)

apple
banana
cherry

[2]: for x in "banana":
    print(x)

b
a
n
a
n
a
```

### *break Statement*

With the **break** statement we can stop the loop before it has looped through all the items:

```
[1]: fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

```
apple
banana
cherry
```

```
[2]: for x in "banana":
    print(x)
```

```
b
a
n
a
n
a
```

```
[3]: fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
    if x == "banana":
        break
```

```
apple
banana
```

```
[4]: |
```

## continue Statement

With the **continue** statement we can stop the current iteration of the loop, and continue with the next:

```
[4]: fruits = ["apple", "banana", "cherry"]
for x in fruits:
    if x == "banana":
        continue
    print(x)
```

```
apple
cherry
```

```
[ ]: |
```

### ***range() Function***

To loop through a set of code a specified number of times, we can use the **range()** function.

The **range()** function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

```
[5]: for x in range(6):
      print(x)
```

```
0
1
2
3
4
5
```

```
[ ]:
```

### ***Else in For Loop***

The **else** keyword in a **for** loop specifies a block of code to be executed when the loop is finished:

```
[6]: for x in range(6):
      print(x)
else:
      print("Finally finished!")
```

```
0
1
2
3
4
5
Finally finished!
```

### ***Nested Loops***

A nested loop is a loop inside a loop.

The "inner loop" will be executed one time for each iteration of the "outer loop":

```
[7]: adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
    for y in fruits:
        print(x, y)
```

```
red apple
red banana
red cherry
big apple
big banana
big cherry
tasty apple
tasty banana
tasty cherry
```

### ***pass Statement***

`for` loops cannot be empty, but if you for some reason have a `for` loop with no content, put in the `pass` statement to avoid getting an error.

```
[8]: for x in [0, 1, 2]:
    pass
```

```
[ ]:
```

## **2.14 Strings & Methods**

Python provides lots of built-in methods which we can use on strings. Below are the list of some string methods available in Python 3.

### **1. `capitalize()`**

Returns a copy of the string with its first character capitalized and the rest lowercased.

```
[13]: mystring = "hello python"
        print(mystring.capitalize())
```

```
Hello python
```

## 2. **CASEFOLD()**

Returns a casefolded copy of the string. Casefolded strings may be used for caseless matching.

```
[14]: mystring = "hello PYTHON"  
      print(mystring.casefold())
```

```
hello python
```

## 3. **Center(width, [fillchar])**

Returns the string centered in a string of length *width*. Padding can be done using the specified *fillchar* (the default padding uses an ASCII space). The original string is returned if *width* is less than or equal to len(s)

```
[15]: mystring = "Hello"  
      x = mystring.center(12,  
                           "-")  
      print(x)
```

```
---Hello----
```

## 4. **Count(sub, [start], [end])**

Returns the number of non-overlapping occurrences of substring (*sub*) in the range [*start*, *end*]. Optional arguments *start* and *end* are interpreted as in slice notation.

```
[18]: mystr = "Hello Python"
print(mystr.count("o"))

print(mystr.count("th"))

print(mystr.count("l"))

print(mystr.count("h"))

print(mystr.count("H"))

print(mystr.count("hH"))
```

```
2
1
2
1
1
0
```

## 5. **Encode(encoding = “utf-8”, errors = “strict”)**

Returns an encoded version of the string as a bytes object. The default encoding is utf-8. errors may be given to set a different error handling scheme. The possible value for errors are:

- Strict (encoding errors raise a unicodeerror)
- Ignore
- Replace
- Xmlcharrefreplace
- Backslashreplace
- Any other name registered via codecs.register\_error()

```
[19]: mystr = 'python!'
print('The string is:',
mystr)
```

The string is: python!

```
[23]: txt = "My name is Ståle"

print(txt.encode(encoding="ascii",errors="backslashreplace"))
print(txt.encode(encoding="ascii",errors="ignore"))
print(txt.encode(encoding="ascii",errors="namereplace"))
print(txt.encode(encoding="ascii",errors="replace"))
print(txt.encode(encoding="ascii",errors="xmlcharrefreplace"))

b'My name is St\\xe5le'
b'My name is Stle'
b'My name is St\\N{LATIN SMALL LETTER A WITH RING ABOVE}le'
b'My name is St?le'
b'My name is St&#229;le'
```

## 6. endswith()

Returns True if the string ends with the specified suffix, otherwise it returns False.

```
[24]: txt = "Hello, welcome to my world."
x = txt.endswith(".")
print(x)
```

True

## 7. upper()

Converts a string into upper case

```
[25]: txt = "Hello my friends"  
       x = txt.upper()  
       print(x)
```

HELLO MY FRIENDS

## 2.15 Formatting in Python

### 1. format() function in Python

The `format()` method has been introduced for handling complex string formatting more efficiently. This method of the built-in string class provides functionality for complex variable substitutions and value formatting. This new formatting technique is regarded as more elegant. The general syntax of `format()`:

```
[1]: txt = "For only {price:.2f} dollars!"  
      print(txt.format(price = 49))
```

For only 49.00 dollars!

### 2. The Placeholders

The placeholders can be identified using named indexes `{price}`, numbered indexes `{0}`, or even empty placeholders `{}`

```
[3]: #named indexes:  
txt1 = "My name is {fname}, I'm {age}.".format(fname = "John", age = 36)  
#numbered indexes:  
txt2 = "My name is {0}, I'm {1}.".format("John", 36)  
#empty placeholders:  
txt3 = "My name is {}, I'm {}.".format("John", 36)  
  
print(txt1)  
print(txt2)  
print(txt3)
```

```
My name is John, I'm 36  
My name is John, I'm 36  
My name is John, I'm 36
```

### 3. Formatting Types

Inside the placeholders you can add a formatting type to format the result:

1. :< Left align

#To demonstrate, we insert the number 8 to set the available space for the value to 8 characters.

#Use "<" to left-align the value:

```
[4]: txt = "We have {:<8} chickens."  
      print(txt.format(49))
```

```
We have 49 chickens.
```

2. :> Right align

```
[5]: txt = "We have {:>8} chickens."  
      print(txt.format(49))
```

```
We have      49 chickens.
```

3. :^ Center align

```
[6]: txt = "We have {:^8} chickens."
      print(txt.format(49))
```

We have 49 chickens.

4. `:=` Places the sign to the left most position

```
[7]: txt = "The temperature is {:=8} degrees celsius."
      print(txt.format(-5))
```

The temperature is - 5 degrees celsius.

5. `:+` To indicate if the result is positive or negative

```
[8]: txt = "The temperature is between {:+} and {:+} degrees celsius."
      print(txt.format(-3, 7))
```

The temperature is between -3 and +7 degrees celsius.

6. `:` Extra space

```
[9]: txt = "The temperature is between {: } and {: } degrees celsius."
      print(txt.format(-3, 7))
```

The temperature is between -3 and 7 degrees celsius.

## 2.16 Python Libraries

Python Libraries are a set of useful functions that eliminate the need for writing codes from scratch.

There are over 137,000 python libraries present today.

Python libraries play a vital role in developing machine learning, data science, data visualization, image and data manipulation applications and more.

### ***What is a Library?***

A library is a collection of pre-combined codes that can be used iteratively to reduce the time required to code. They are particularly useful for accessing the pre-written frequently used codes, instead of writing them from scratch every single time. Similar to the physical libraries, these are a collection of reusable resources, which means

every library has a root source. This is the foundation behind the numerous open-source libraries available in Python.

## Some basic Python Libraries

### 1. Scikit- learn

It is a free software machine learning library for the Python programming language and can be effectively used for a variety of applications which include classification, regression, clustering, model selection, naive Bayes', grade boosting, K-means, and pre-processing.

### 2. NumPy

NumPy is one of the fundamental packages for Python providing support for large multidimensional arrays and matrices along with a collection of high-level mathematical functions to execute these functions swiftly. NumPy relies on BLAS and LAPACK for efficient linear algebra computations. NumPy can also be used as an efficient multi-dimensional container of generic data.

### 3. TensorFlow

The most popular deep learning framework, TensorFlow is an open-source software library for high-performance numerical computation. It is an iconic math library and is also used for machine learning and deep learning algorithms. Tensorflow was developed by the researchers at the Google Brain team within Google AI organisation, and today it is being used by researchers for machine learning algorithms, and by physicists for complex mathematical computations.

### 4. Keras

It is an open-source neural network library written in Python designed to enable fast experimentation with deep neural networks. With deep learning becoming ubiquitous, Keras becomes the ideal choice as it is API designed for humans and not machines according to the creators.

### 5. Pandas

It is an open-source, BSD licensed library. Pandas enable the provision of easy data structure and quicker data analysis for Python. For operations like data analysis and modelling, Pandas makes it possible to carry these out without needing to switch to more domain-specific language like R.

### 6. OpenCV Python

Open Source Computer Vision or OpenCV is used for image processing. It is a Python package that monitors overall functions focused on instant computer vision. OpenCV provides several inbuilt functions, with the help of this you can learn Computer Vision. It allows both read and write images at the same time. Objects such as faces, trees, etc., can be diagnosed in any video or image.

After understanding the predefined libraries lets learn how to create our own library and how to use it.

## 2.17 User-Defined Library

Creating a user defined module.

1. To create a module just save the code you want in a file with the file extension .py:

Example;

Save this code in a file named `mymodule.py`

```
def greeting(name):
    print("Hello, " + name)
```

### 2. Using the Module

Now we can use the module we just created, by using the import statement:

Example

Import the module named mymodule, and call the greeting function:

```
import mymodule
```

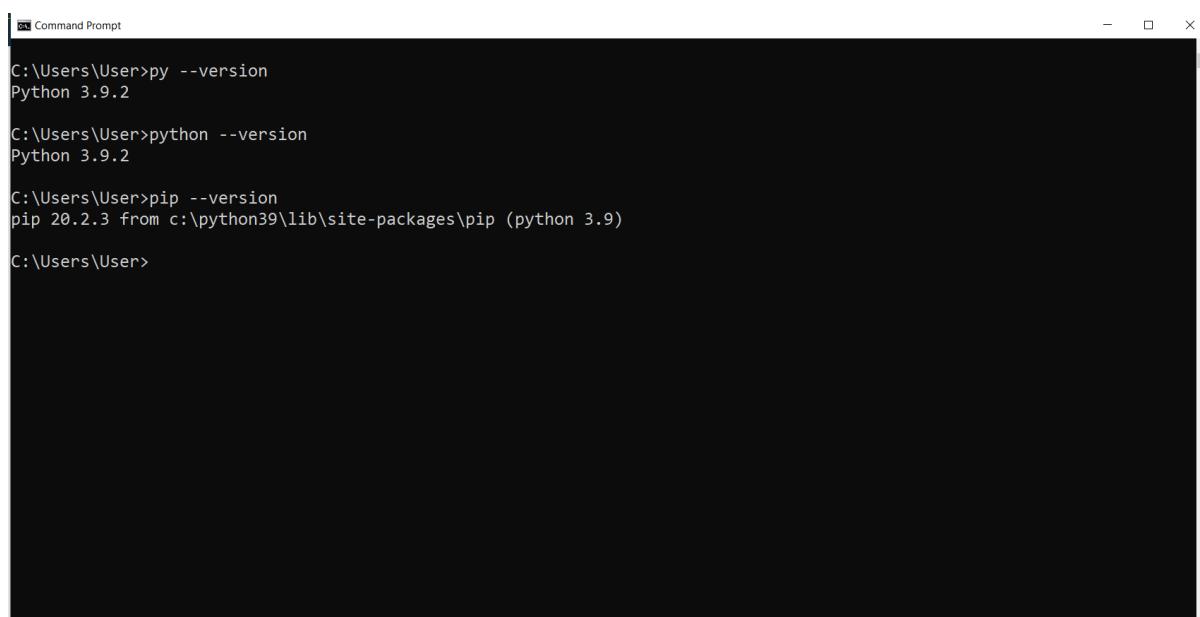
```
mymodule.greeting("Jonathan")
```

## Installing Packages with pip

`pip` is the standard package manager for [Python](#). It allows you to install and manage additional packages that are not part of the [Python standard library](#).

1. Check, whether you have an installed version

```
py --version
pip --version
```



```
Command Prompt
C:\Users\User>py --version
Python 3.9.2

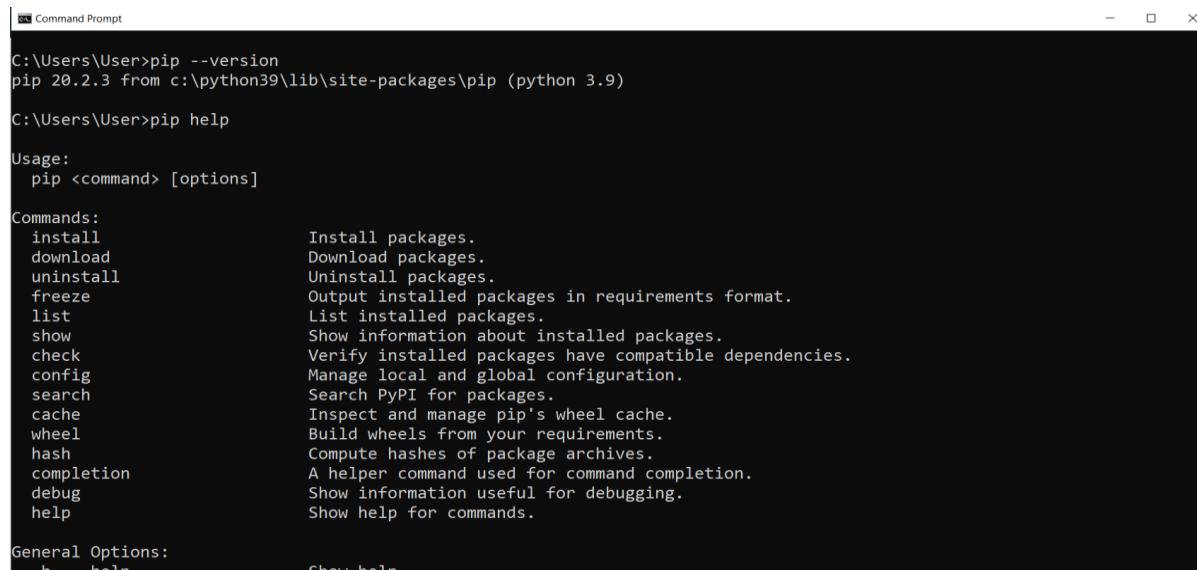
C:\Users\User>python --version
Python 3.9.2

C:\Users\User>pip --version
pip 20.2.3 from c:\python39\lib\site-packages\pip (python 3.9)

C:\Users\User>
```

2. If you need help, please try help

## pip help



```
C:\Users\User>pip --version
pip 20.2.3 from c:\python39\lib\site-packages\pip (python 3.9)

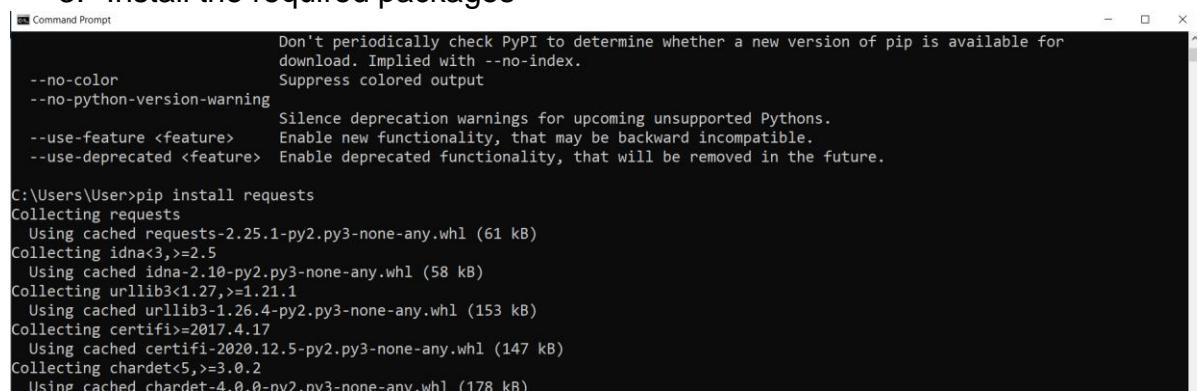
C:\Users\User>pip help

Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements format.
  list               List installed packages.
  show               Show information about installed packages.
  check              Verify installed packages have compatible dependencies.
  config             Manage local and global configuration.
  search             Search PyPI for packages.
  cache              Inspect and manage pip's wheel cache.
  wheel              Build wheels from your requirements.
  hash               Compute hashes of package archives.
  completion         A helper command used for command completion.
  debug              Show information useful for debugging.
  help               Show help for commands.

General Options:
  -h, --help          Show help.
```

### 3. Install the required packages



```
Don't periodically check PyPI to determine whether a new version of pip is available for
download. Implied with --no-index.
--no-color           Suppress colored output
--no-python-version-warning
--use-feature <feature>   Silence deprecation warnings for upcoming unsupported Pythons.
--use-deprecated <feature> Enable new functionality, that may be backward incompatible.
                           Enable deprecated functionality, that will be removed in the future.

C:\Users\User>pip install requests
Collecting requests
  Using cached requests-2.25.1-py2.py3-none-any.whl (61 kB)
Collecting idna<3,>=2.5
  Using cached idna-2.10-py2.py3-none-any.whl (58 kB)
Collecting urllib3<1.27,>=1.21
  Using cached urllib3-1.26.4-py2.py3-none-any.whl (153 kB)
Collecting certifi==2017.4.17
  Using cached certifi-2020.12.5-py2.py3-none-any.whl (147 kB)
Collecting chardet<5,>=3.0.2
  Using cached chardet-4.0.0-py2.py3-none-any.whl (178 kB)
```

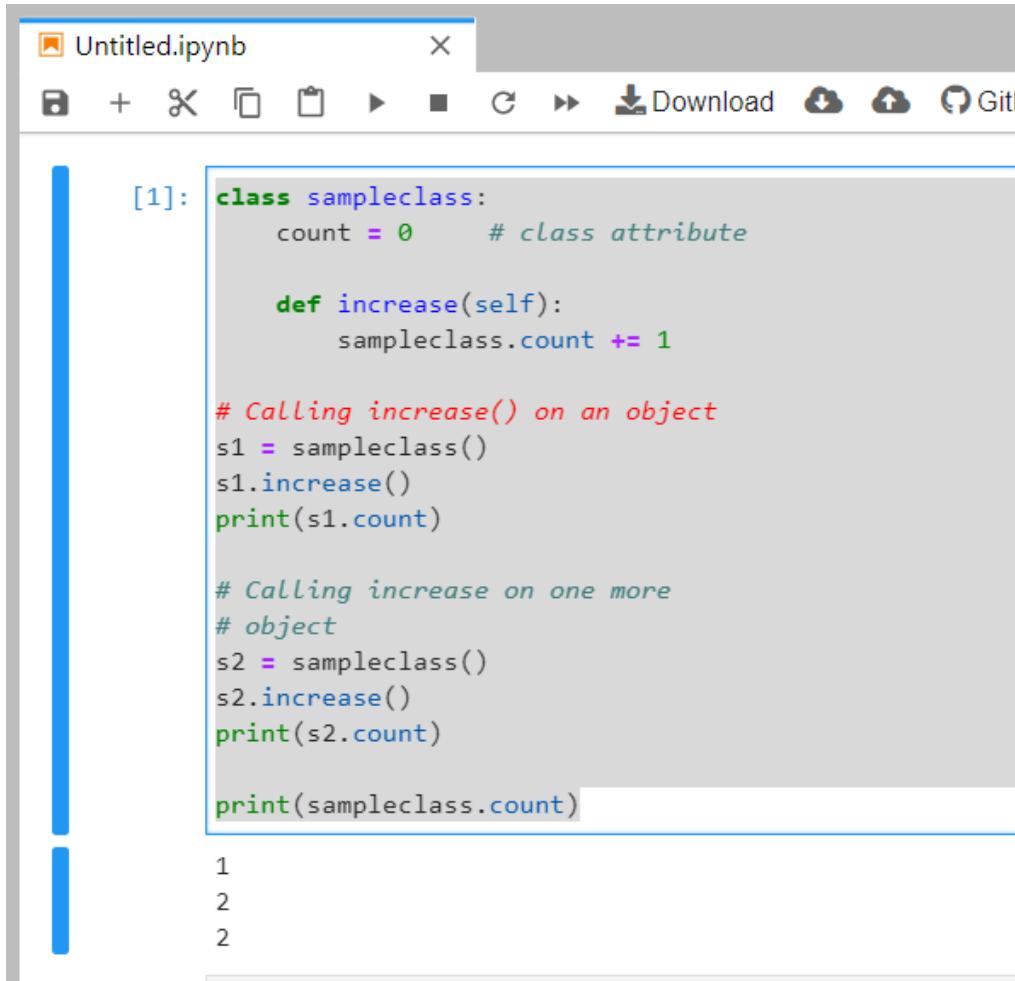
Let's see the object-oriented feature of python- class

An object is simply a collection of data (variables) and methods (functions) that act on those data. Similarly, a class is a blueprint for that object.

We can think of class as a sketch (prototype) of a house. It contains all the details about the floors, doors, windows etc. Based on these descriptions we build the house. House is the object.

## Attributes

Class attributes belong to the class itself they will be shared by all the instances. Such attributes are defined in the class body parts usually at the top, for legibility.



```
[1]: class sampleclass:
    count = 0      # class attribute

    def increase(self):
        sampleclass.count += 1

# Calling increase() on an object
s1 = sampleclass()
s1.increase()
print(s1.count)

# Calling increase on one more
# object
s2 = sampleclass()
s2.increase()
print(s2.count)

print(sampleclass.count)
```

1  
2  
2

## Instance Attributes

Unlike class attributes, instance attributes are not shared by objects. Every object has its own copy of the instance attribute (In case of class attributes all object refer to single copy).

To list the attributes of an instance/object, we have two functions:-  
1. **vars()**- This function displays the attribute of an instance in the form of an dictionary.

2. **dir()**- This function displays more attributes than vars function, as it is not limited to instance. It displays the class attributes as well. It also displays the attributes of its ancestor classes.

So, in this chapter we explored the fundamentals of Linux operating system and basics of Python programming language. Happy Learning.

Please refer this [link \[Click Here\]](#) for all the codes/practical in this chapter.

# Chapter 3: Data Analysis with Python

## Learning Outcomes:

- Understand the statistical concept using Numpy Library
- Use Matplotlib library and its various functions to visualize the data
- Analyse different types of data using Pandas
- Understand the use of data analytics in improved decision making

Let us now get started with libraries which are required during Data Analysis like Numpy, Pandas and Matplotlib.

## 3.1 Numpy Library

NumPy is an open source Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy stands for Numerical Python.

### *Why NumPy?*

In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called `ndarray`, it provides a lot of supporting functions that make working with `ndarray` very easy. Arrays are very frequently used in data science, where speed and resources are very important.

Numpy arrays are stored in a single contiguous (continuous) block of memory. There are two key concepts relating to memory: dimensions and **strides**.

Firstly, many Numpy functions use strides to make things fast. Examples include integer slicing (e.g. `X[1,0:2]`) and broadcasting. Understanding strides helps us better understand how Numpy operates.

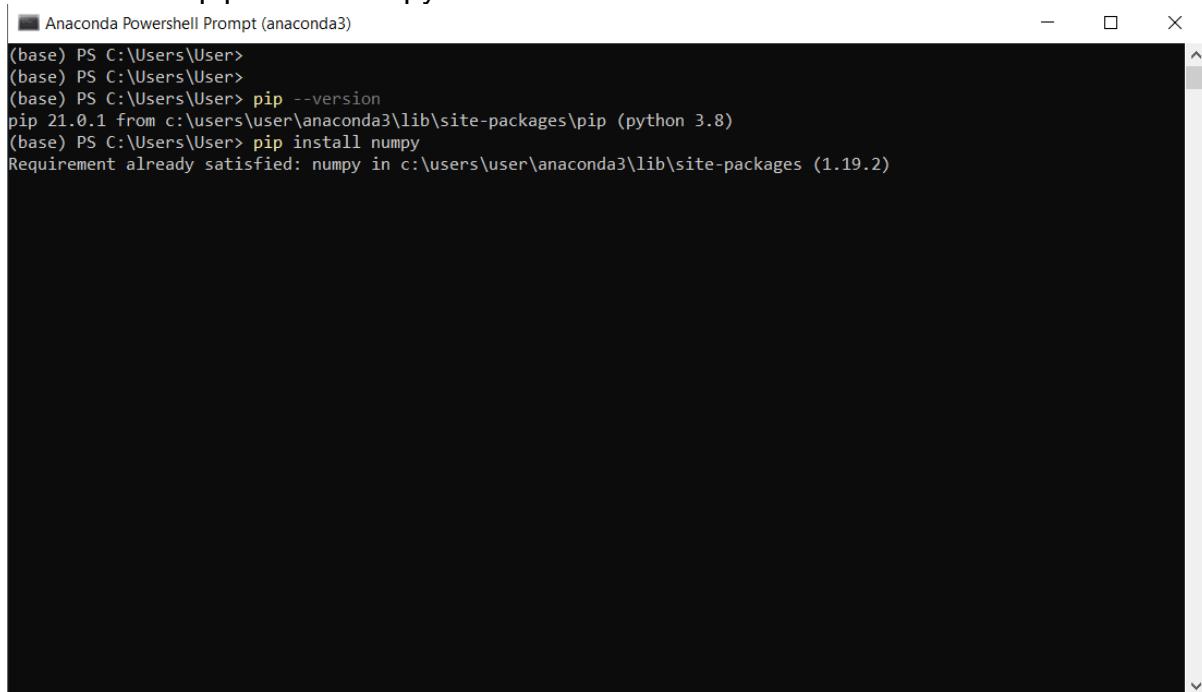
Secondly, we can directly use strides to make our own code faster. This can be particularly useful for data pre-processing in machine learning.

NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

### ***Installing Numpy Module***

1. To install numpy library,

```
pip install numpy
```



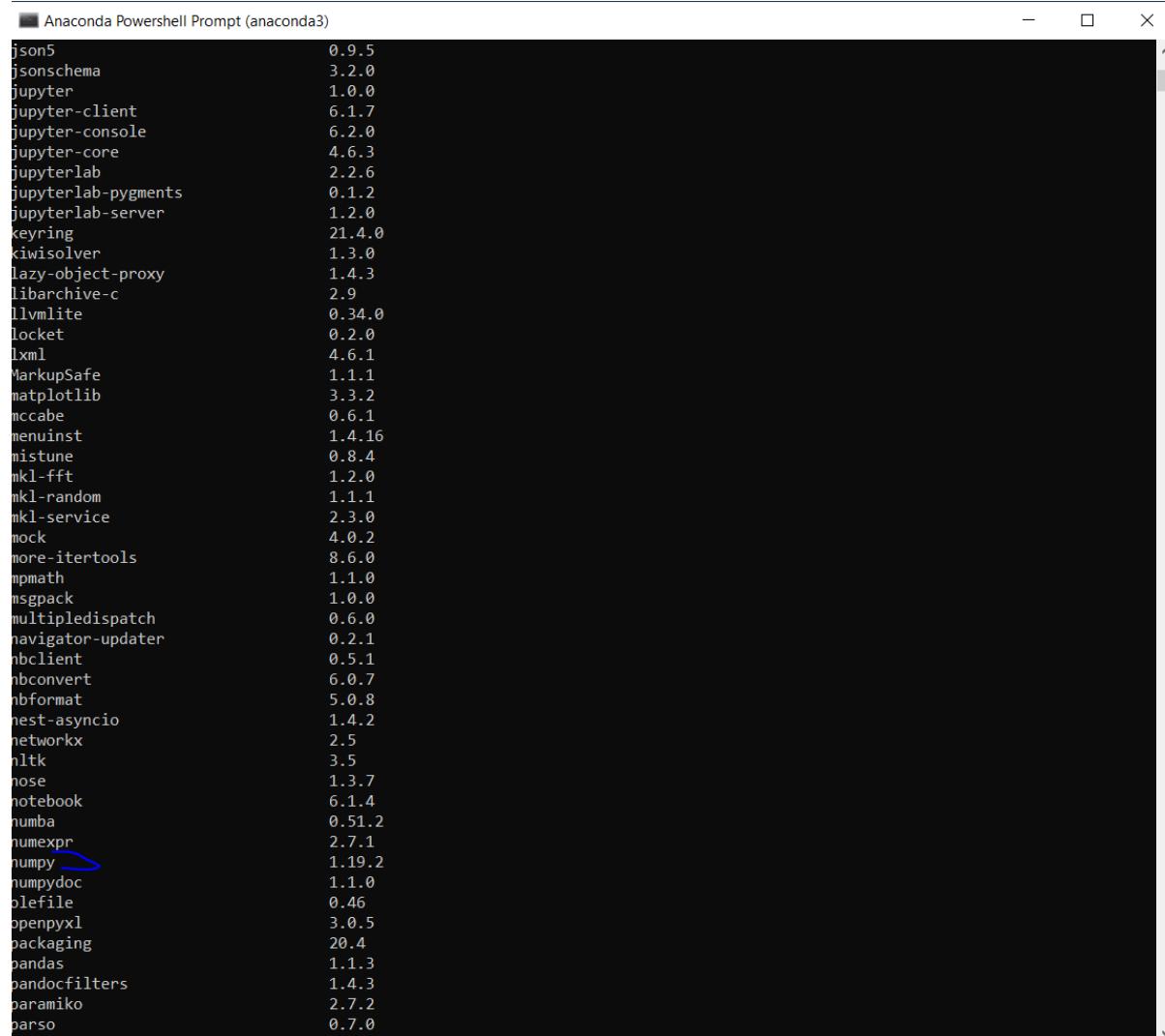
The screenshot shows a terminal window titled "Anaconda Powershell Prompt (anaconda3)". The command "pip install numpy" is entered and executed. The output shows that pip 21.0.1 is used from the anaconda3 library, and numpy version 1.19.2 is already installed.

```
[base] PS C:\Users\User>
(base) PS C:\Users\User>
(base) PS C:\Users\User> pip --version
pip 21.0.1 from c:\users\user\anaconda3\lib\site-packages\pip (python 3.8)
(base) PS C:\Users\User> pip install numpy
Requirement already satisfied: numpy in c:\users\user\anaconda3\lib\site-packages (1.19.2)
```

2. To verify the libraries already installed,

```
pip list
```

```
■ Anaconda Powershell Prompt (anaconda3)
(base) PS C:\Users\User>
(base) PS C:\Users\User> pip --version
pip 21.0.1 from c:\users\user\anaconda3\lib\site-packages\pip (python 3.8)
(base) PS C:\Users\User> pip install numpy
Requirement already satisfied: numpy in c:\users\user\anaconda3\lib\site-packages (1.19.2)
(base) PS C:\Users\User> pip list
Package           Version
-----
alabaster        0.7.12
anaconda-client   1.7.2
anaconda-navigator 1.10.0
anaconda-project  0.8.3
argh              0.26.2
argon2-cffi       20.1.0
asn1crypto        1.4.0
astroid            2.4.2
astropy            4.0.2
async-generator    1.10
atomicwrites      1.4.0
attrs              20.3.0
autopenp8          1.5.4
Babel              2.8.1
backcall            0.2.0
backports.functools-lru-cache 1.6.3
backports.shutil-get-terminal-size 1.0.0
backports.tempfile    1.0
backports.weakref     1.0.post1
bcrypt              3.2.0
beautifulsoup4      4.9.3
bitarray            1.6.1
bkcharts             0.2
bleach              3.2.1
bokeh              2.2.3
boto                2.49.0
Bottleneck          1.3.2
brotliipy          0.7.0
certifi            2020.6.20
cffi                1.14.3
chardet            3.0.4
click               7.1.2
cloudpickle         1.6.0
clyent              1.2.2
colorama            0.4.4
comtypes            1.1.7
conda              4.10.1
conda-build          3.20.5
conda-package-handling 1.7.3
conda-verify         3.4.2
```



Package	Version
json5	0.9.5
jsonschema	3.2.0
jupyter	1.0.0
jupyter-client	6.1.7
jupyter-console	6.2.0
jupyter-core	4.6.3
jupyterlab	2.2.6
jupyterlab-pygments	0.1.2
jupyterlab-server	1.2.0
keyring	21.4.0
kiwisolver	1.3.0
lazy-object-proxy	1.4.3
libarchive-c	2.9
llvmlite	0.34.0
locket	0.2.0
lxml	4.6.1
MarkupSafe	1.1.1
matplotlib	3.3.2
mccabe	0.6.1
menuinst	1.4.16
mistune	0.8.4
mkl-fft	1.2.0
mkl-random	1.1.1
mkl-service	2.3.0
mock	4.0.2
more-itertools	8.6.0
mpmath	1.1.0
msgpack	1.0.0
multipledispatch	0.6.0
navigator-updater	0.2.1
nbclient	0.5.1
nbconvert	6.0.7
nbformat	5.0.8
nest-asyncio	1.4.2
networkx	2.5
nltk	3.5
nose	1.3.7
notebook	6.1.4
numba	0.51.2
numexpr	2.7.1
numpy	1.19.2
numpydoc	1.1.0
olefile	0.46
openpyxl	3.0.5
packaging	20.4
pandas	1.1.3
pandocfilters	1.4.3
paramiko	2.7.2
parso	0.7.0

## Create a NumPy ndarray Object

NumPy is used to work with arrays. The array object in NumPy is called `ndarray`. We can create a NumPy `ndarray` object by using the `array()` function.

### 1. Create a NumPy ndarray Object

NumPy is used to work with arrays. The array object in NumPy is called `ndarray`. We can create a NumPy `ndarray` object by using the `array()` function.

```
In [3]: import numpy as np  
  
arr = np.array([1, 2, 3, 4, 5])  
  
print(arr)  
  
print(type(arr))
```

```
[1 2 3 4 5]  
<class 'numpy.ndarray'>
```

### *Min, Max & Everything in between*

```
In [4]: # import numpy library  
import numpy  
  
# creating a numpy array of integers  
arr = numpy.array([1, 5, 4, 8, 3, 7])  
  
# finding the maximum and  
# minimum element in the array  
max_element = numpy.max(arr)  
min_element = numpy.min(arr)  
  
# printing the result  
print('maximum element in the array is: ',  
     max_element)  
print('minimum element in the array is: ',  
     min_element)
```

```
maximum element in the array is: 8  
minimum element in the array is: 1
```

## Random Number Generator

### 1. Generate Random Number

NumPy offers the random module to work with random numbers.

Ex: Generate a random integer from 0 to 100 :

```
In [5]: from numpy import random  
  
x = random.randint(100)  
  
print(x)
```

89

```
In [6]: from numpy import random  
  
x = random.randint(100)  
  
print(x)
```

45

## 2. Generate Random Float

The random module's `rand()` method returns a random float between 0 and 1.

Ex: Generate a random float from 0 to 1 :

```
In [7]: from numpy import random  
  
x = random.rand()  
  
print(x)
```

0.43203375825754287

## 3. Generate Random Array

In NumPy we work with arrays, and you can use the two methods from the above examples to make random arrays.

### Integers

The `randint()` method takes a size parameter where you can specify the shape of an array.

Ex: Generate a 1-D array containing 5 random integers from 0 to 100

```
In [8]: from numpy import random  
  
x=random.randint(100, size=(5))  
  
print(x)  
  
[71 53 93 89 47]
```

Ex: Generate a 2-D array with 3 rows, each row containing 5 random integers from 0 to 100

```
In [9]: from numpy import random  
  
x = random.randint(100, size=(3, 5))  
  
print(x)  
  
[[75  8 37 45  4]  
 [39 17 51 75 29]  
 [66 61 48 52 83]]
```

## 3.2 Creating scalars in numpy

```
In [13]: # Python program explaining  
# numpy.asscalar() function  
  
import numpy as edunet  
# creating a array of size 1  
in_arr = edunet.array([ 8 ])  
  
print ("Input array : ", in_arr)  
  
→  
out_scalar = edunet.asscalar(in_arr)  
print ("output scalar from input array : ", out_scalar)
```

```
Input array : [8]  
output scalar from input array : 8
```

### 3.3 Creating Vector in numpy

```
In [12]: # importing numpy
import numpy as np

# creating a 1-D list (Horizontal)
list1 = [1, 2, 3]

# creating a 1-D list (Vertical)
list2 = [[10],
          [20],
          [30]]

# creating a vector1
# vector as row
vector1 = np.array(list1)

# creating a vector 2
# vector as column
vector2 = np.array(list2)

# showing horizontal vector
print("Horizontal Vector")
print(vector1)

print("-----")

# showing vertical vector
print("Vertical Vector")
print(vector2)
```

```
Horizontal Vector
[1 2 3]
-----
Vertical Vector
[[10]
 [20]
 [30]]
```

### 3.4 Creating Matrix in numpy

```
# Python Program illustrating
# numpy.matrix class

import numpy as edunet

# string input
a = edunet.matrix('1 2; 3 4')
print("Via string input : \n", a, "\n\n")

# array-like input
b = edunet.matrix([[5, 6, 7], [4, 6]])
print("Via array-like input : \n", b)
```

Via string input :  
[[1 2]  
[3 4]]

Via array-like input :  
[[list([5, 6, 7]) list([4, 6])]]

### 3.5 Matrix Multiplication in Numpy

Matrix multiplication is an operation that takes two matrices as input and produces single matrix by multiplying rows of the first matrix to the column of the second matrix. In matrix multiplication make sure that the number of rows of the first matrix should be equal to the number of columns of the second matrix.

Ex;

```
Input:matrix1 = ([1, 2, 3],
                 [3, 4, 5],
                 [7, 6, 4])
matrix2 = ([5, 2, 6],
           [5, 6, 7],
           [7, 6, 4])
```

```
Output : [[36 32 32]
          [70 60 66]
          [93 74 100]]
```

```
In [14]: import numpy as np

# input two matrices
mat1 = ([1, 6, 5],[3 ,4, 8],[2, 12, 3])
mat2 = ([3, 4, 6],[5, 6, 7],[6,56, 7])

# This will return dot product
res = np.dot(mat1,mat2)

# print resulted matrix
print(res)
```

```
[[ 63 320 83]
 [ 77 484 102]
 [ 84 248 117]]
```

## 3.6 NumPy Statistical Functions

Statistics is concerned with collecting and then analyzing that data. It includes methods for collecting the samples, describing the data, and then concluding that data. NumPy is the fundamental package for scientific calculations and hence goes hand-in-hand for NumPy statistical Functions.

NumPy contains various statistical functions that are used to perform statistical data analysis. These statistical functions are useful when finding a maximum or minimum of elements. It is also used to find basic statistical concepts like standard deviation, variance, etc.

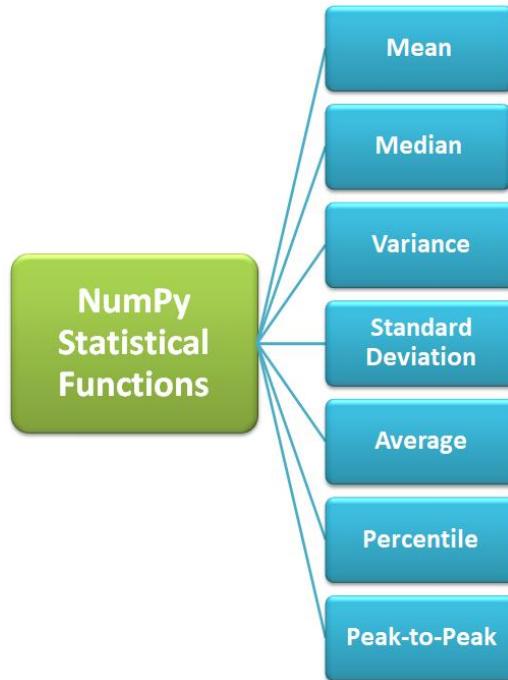


Fig: Numpy Statistical Functions

## NumPy Statistical Functions

NumPy is equipped with the following statistical functions, Let us see one by one:

1. `np.amin()`- This function determines the minimum value of the element along a specified axis.
2. `np.amax()`- This function determines the maximum value of the element along a specified axis.
3. `np.mean()`- It determines the mean value of the data set.
4. `np.median()`- It determines the median value of the data set.
5. `np.std()`- It determines the standard deviation
6. `np.var()` – It determines the variance.
7. `np.ptp()`- It returns a range of values along an axis.
8. `np.average()`- It determines the weighted average
9. `np.percentile()`- It determines the nth percentile of data along the specified axis.

### 1. Finding maximum and minimum of array in NumPy

NumPy `np.amin()`and `np.amax()`functions are useful to determine the minimum and maximum value of array elements along a specified axis.

```
In [17]: import numpy as np
arr= np.array([[1,23,78],[98,60,75],[79,25,48]])
print(arr)
#Minimum Function
print(np.amin(arr))
#Maximum Function
print(np.amax(arr))
```

```
[[ 1 23 78]
 [98 60 75]
 [79 25 48]]
1
98
```

## 2. Finding Mean, Median, Standard Deviation and Variance in NumPy

### Mean

Mean is the sum of the elements divided by the number of elements and is given by the following formula:

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \dots + x_n)$$

It calculates the mean by adding all the items of the arrays and then divides it by the number of elements. We can also mention the axis along which the mean can be calculated.

```
In [18]: import numpy as np
a = np.array([5,6,7])
print(a)
print(np.mean(a))
```

```
[5 6 7]
6.0
```

### Median

Median is the middle element of the array. The formula differs for odd and even sets. The median for a given set of data with n elements (observations) is given by

$$\text{Odd data} = \left(\frac{n+1}{2}\right)^{\text{th}} \text{ observation}, \quad \text{Even data} = \frac{\frac{n}{2}^{\text{th}} \text{ obs.} + \left(\frac{n}{2}+1\right)^{\text{th}} \text{ obs.}}{2}$$

It can calculate the median for both one-dimensional and multi-dimensional arrays. Median separates the higher and lower range of data values.

```
In [19]: import numpy as np  
a = np.array([5,6,7])  
print(a)  
print(np.median(a))
```

```
[5 6 7]  
6.0
```

### **Standard Deviation**

Standard deviation is the square root of the average of square deviations from mean. The formula for standard deviation is:

$$SD = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}}$$

```
In [20]: import numpy as np  
a = np.array([5,6,7])  
print(a)  
print(np.std(a))
```

```
[5 6 7]  
0.816496580927726
```

---

### **Variance**

Variance is the average of the squared differences from the mean. Following is the formula for the same:

$$\sigma^2 = \frac{\sum (x_r - \bar{x})^2}{n}$$

```
In [21]: import numpy as np
a = np.array([5,6,7])
print(a)
print(np.var(a))
```

```
[5 6 7]
0.6666666666666666
```

## 3.7 Percentile & Interquartile in Numpy

*Quartiles :*

A quartile is a type of quantile. The first quartile (Q1), is defined as the middle number between the smallest number and the median of the data set, the second quartile (Q2) – median of the given data set while the third quartile (Q3), is the middle number between the median and the largest value of the data set.

*Algorithm to find Quartiles:*

Quartiles are calculated by the help of the median. If the number of entries is an even number i.e. of the form  $2n$ , then, first quartile (Q1) is equal to the median of the  $n$  smallest entries and the third quartile (Q3) is equal to the median of the  $n$  largest entries.

If the number of entries is an odd number i.e. of the form  $(2n + 1)$ , then

- the first quartile (Q1) is equal to the median of the  $n$  smallest entries
- the third quartile (Q3) is equal to the median of the  $n$  largest entries
- the second quartile(Q2) is the same as the ordinary median.

*Range:* It is the difference between the largest value and the smallest value in the given data set.

*Interquartile Range :* The interquartile range (IQR), also called as midspread or middle 50%, or technically H-spread is the difference between the third quartile (Q3) and the first quartile (Q1). It covers the center of the distribution and contains 50% of the observations.  $IQR = Q3 - Q1$

Uses;

- The interquartile range has a breakdown point of 25% due to which it is often preferred over the total range.
- The IQR is used to build box plots, simple graphical representations of a probability distribution.
- The IQR can also be used to identify the outliers in the given data set.
- The IQR gives the central tendency of the data.

## Decision Making

- The data set having a higher value of interquartile range (IQR) has more variability.
- The data set having a lower value of interquartile range (IQR) is preferable.

### Interquartile range using numpy.median

```
In [22]: import numpy as np

data = [32, 36, 46, 47, 56, 69, 75, 79, 79, 88, 89, 91, 92, 93, 96, 97,
       101, 105, 112, 116]

# First quartile (Q1)
Q1 = np.median(data[:10])

# Third quartile (Q3)
Q3 = np.median(data[10:])

# Interquartile range (IQR)
IQR = Q3 - Q1

print(IQR)
```

34.0

### Interquartile range using numpy.percentile

```
In [23]: # Import numpy library
import numpy as np

data = [32, 36, 46, 47, 56, 69, 75, 79, 79, 88, 89, 91, 92, 93, 96, 97,
       101, 105, 112, 116]

# First quartile (Q1)
Q1 = np.percentile(data, 25, interpolation = 'midpoint')

# Third quartile (Q3)
Q3 = np.percentile(data, 75, interpolation = 'midpoint')

# Interquartile range (IQR)
IQR = Q3 - Q1

print(IQR)
```

34.0

After knowing the various statistical functions in NumPy which are used in descriptive analytics, let us see some other interesting functionalities of NumPy.

## 3.8 Array Broadcasting in Numpy

The term broadcasting describes how numpy treats arrays with different shapes during arithmetic operations. Subject to certain constraints, the smaller array is “broadcast” across the larger array so that they have compatible shapes. [ 4.9 ]

```
In [27]: a = np.array([1.0, 2.0, 3.0])
b = np.array([2.0, 2.0, 2.0])
a * b
```

Out[27]: array([2., 4., 6.])

```
In [28]: from numpy import array
a = array([1.0, 2.0, 3.0])
b = array([2.0, 2.0, 2.0])
a * b
```

Out[28]: array([2., 4., 6.])

---

```
In [29]: from numpy import array
a = array([1.0, 2.0, 3.0])
b = 2.0
a * b
```

Out[29]: array([2., 4., 6.])

---

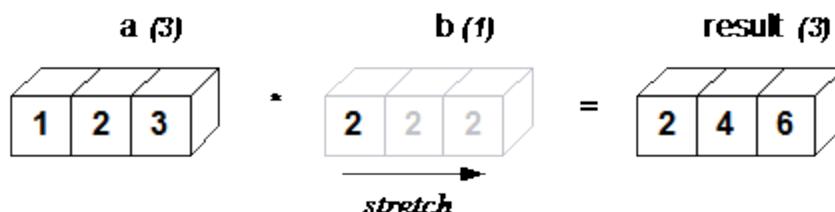


Figure 1: Broadcast Array

Ref: <https://numpy.org/devdocs/user/theory.broadcasting.html#figure-1>

*In the simplest example of broadcasting, the scalar ‘‘b’’ is stretched to become an array of same shape as ‘‘a’’ so the shapes are compatible for element-by-element multiplication.*

The result is equivalent to the previous example where b was an array. We can think of the scalar b being stretched during the arithmetic operation into an array with the

same shape as a. The new elements in b, as shown in Figure 1, are simply copies of the original scalar.

### Sorting Arrays

`numpy.sort()` : This function returns a sorted copy of an array.

Parameters;

`arr` : Array to be sorted.

`axis` : Axis along which we need array to be started.

`order` : This argument specifies which fields to compare first.

`kind` : ['quicksort'{default}, 'mergesort', 'heapsort']Sorting algorithm.

```
In [30]: import numpy as np

# sort along the first axis
a = np.array([[12, 15], [10, 1]])
arr1 = np.sort(a, axis = 0)
print ("Along first axis : \n", arr1)

# sort along the last axis
a = np.array([[10, 15], [12, 1]])
arr2 = np.sort(a, axis = -1)
print ("\nAlong first axis : \n", arr2)

a = np.array([[12, 15], [10, 1]])
arr1 = np.sort(a, axis = None)
print ("\nAlong none axis : \n", arr1)

Along first axis :
[[10  1]
 [12 15]]

Along first axis :
[[10 15]
 [ 1 12]]

Along none axis :
[ 1 10 12 15]
```

## 3.9 Data Creation

The Pandas library is one of the most preferred tools for data scientists to do data manipulation and analysis, next to matplotlib for data visualization and NumPy, the fundamental library for scientific computing in Python on which Pandas was built.

The fast, flexible, and expressive Pandas data structures are designed to make real-world data analysis significantly easier, but this might not be immediately the case for those who are just getting started with it. This is because there is so much functionality built into this package that the options are overwhelming.

Meanwhile, Pandas will be introduced at a later stage. Concluding, we have discussed Data Creation, with arrays & matrices using Numpy through this module.

### 3.10 What is Data Visualization?

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

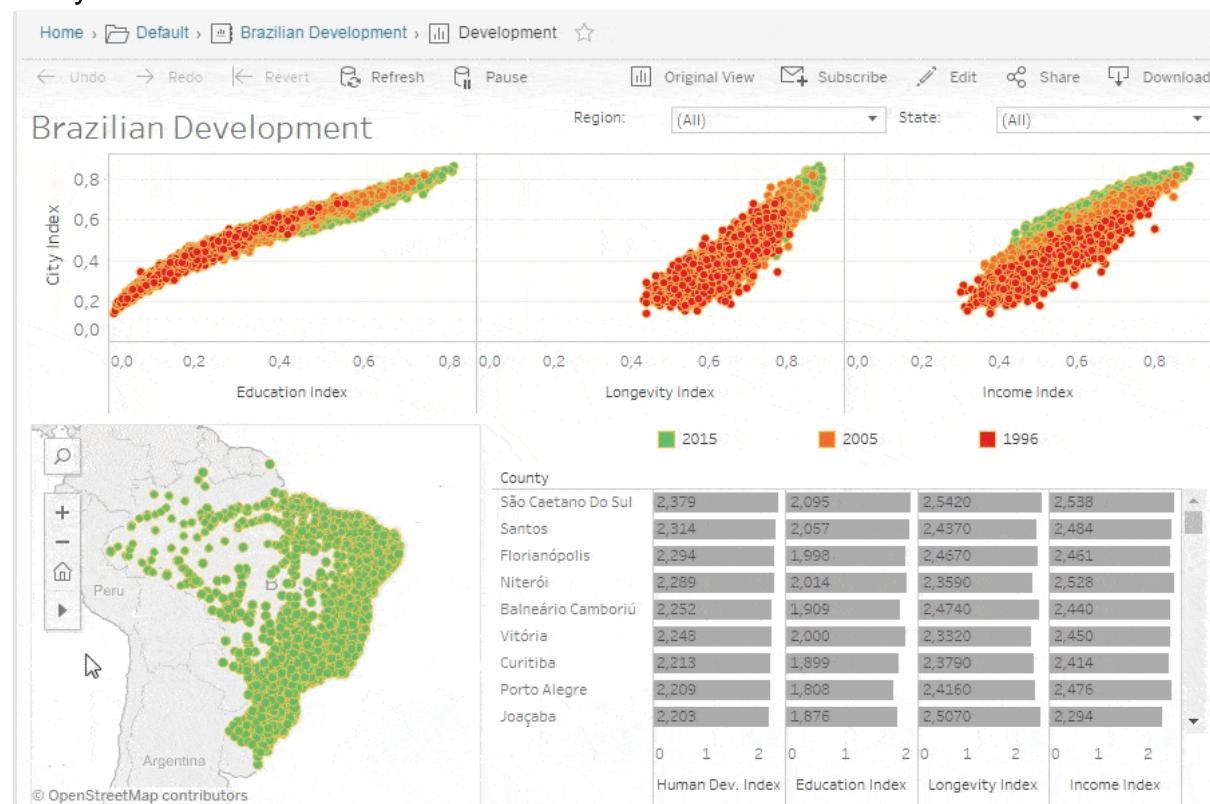


Image: sample Data visualization

Reference: [https://cdn1.tblsft.com/sites/default/files/pages/\\_data\\_visualization\\_definition.gif](https://cdn1.tblsft.com/sites/default/files/pages/_data_visualization_definition.gif)

Our eyes are drawn to colors and patterns. We can quickly identify red from blue, square from circle. Our culture is visual, including everything from art and advertisements to TV and movies.

Data visualization is another form of visual art that grabs our interest and keeps our eyes on the message. When we see a chart, we quickly see trends and outliers. If we can see something, we internalize it quickly. It's storytelling with a purpose. If you've ever stared at a massive spreadsheet of data and couldn't see a trend, you know how much more effective a visualization can be.

Data visualization helps transform your data into an engaging story with details and patterns.

- Better Analysis
- Speed up decision making process
- Quick action
- Identifying patterns
- Story telling is more engaging
- Grasping the latest trends
- Finding errors

## Families of Visualizations

Data Visualization's Family speaks to its nature. This allows for hierarchical classification between major groups: Charts, Geospatial Visualizations, and Tables.

### Charts

A chart is a representation of data in the form of a graph, diagram, map, or tabular format. This could make the other two families, Geospatial and Tables, subfamilies of it.

Later, we will discuss about different types of charts in detail.

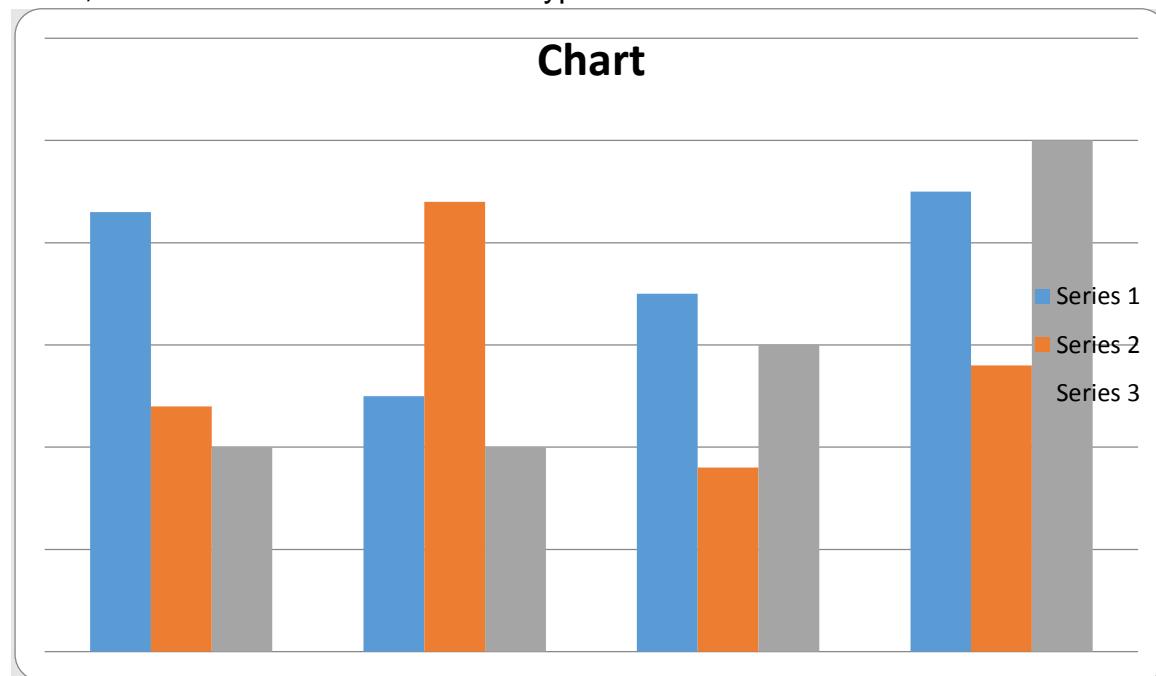


Image: Sample Chart

## Geospatial visualization or Geovisualization

These visualizations focus on the relationship between data and its location to create insight. Any positional data works for spatial analysis.



Image: sample Geospatial visualization

Reference: <https://www.tableau.com/learn/articles/data-visualization/glossary>

## Tables

Tables, also known as “crosstabs” or “spreadsheets”, are one of the most common ways to present data. They may not be visualizations in themselves, but they can be a powerful tool for visual analysis

1234	678
368	8034
2620	2559
971	322

Image: sample tables visualization

Reference: <https://www.tableau.com/learn/articles/data-visualization/glossary>

## Plotting and Visualization

*Plotting* is a chart or map showing the movements or progress of an object.

A plot is a graphical technique for representing a data set, usually as a graph showing the relationship between two or more variables.

Python offers multiple great graphing libraries that come packed with lots of different features.

## To get a little overview here are a few popular plotting libraries:

- [Matplotlib](#): low level, provides lots of freedom
- [Pandas Visualization](#): easy to use interface, built on Matplotlib
- [Seaborn](#): high-level interface, great default styles
- [ggplot](#): based on R's ggplot2, uses Grammar of Graphics
- [Plotly](#): can create interactive plots

Let us explore the Matplotlib library.

## Matplotlib

Here we will learn how to create basic plots using Matplotlib.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

### How to install matplotlib in python?

```
pip install matplotlib  
or  
conda install matplotlib
```

```
import matplotlib.pyplot as plt
```

`matplotlib.pyplot` is a collection of command style functions that make matplotlib work like MATLAB.

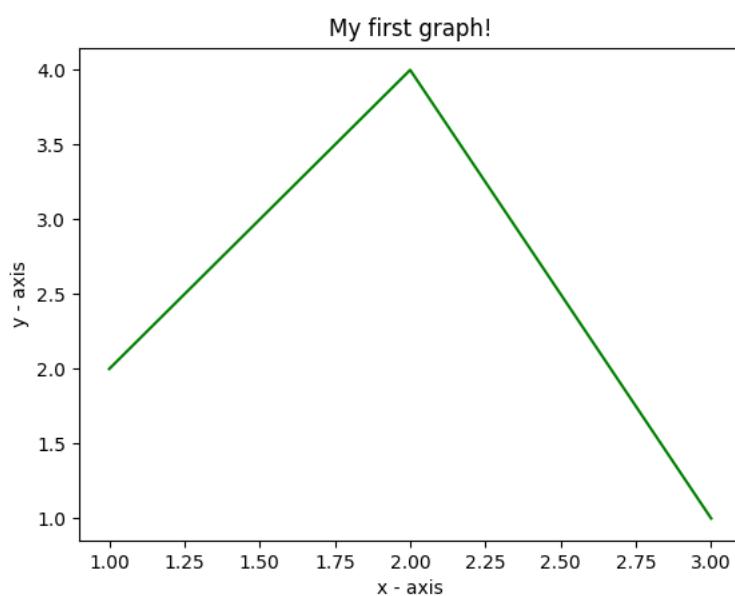
### 3.11 Graph Plotting in Python using matplotlib

In [1]:

```
# importing the required module
import matplotlib.pyplot as plt
# x axis values
x = [1,2,3]
# corresponding y axis values
y = [2,4,1]
# plotting the points
plt.plot(x, y)
# naming the x axis
plt.xlabel('x - axis')
# naming the y axis
plt.ylabel('y - axis')
# giving a title to my graph
plt.title('My first graph!')
# function to show the plot
plt.show()
```

Code: Graph Plotting in Python using matplotlib.ipynb

#### Output:



The code seems self explanatory. Following steps were followed:

- Define the x-axis and corresponding y-axis values as lists.
- Plot them on canvas using .plot() function.
- Give a name to x-axis and y-axis using .xlabel() and .ylabel() functions.
- Give a title to your plot using .title() function.
- Finally, to view your plot, we use .show() function.

## 3.12 Figures and subplots

### Figures in matplotlib

The **Figure** is the overall window or page on which everything is drawn on. The `matplotlib.figure` module contains the Figure class. It is a top-level container for all plot elements. The Figure object is instantiated by calling the `figure()` function from the `pyplot` module –

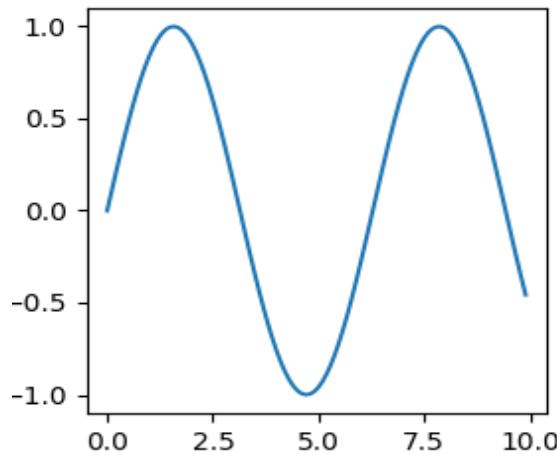
Syntax : `fig = plt.figure()`

```
In [1]: import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0, 10, 0.1)
y = np.sin(x)

# figures in matplotlib using figure()
plt.figure(figsize=(3, 3))
plt.plot(x, y)
plt.show()
```

Code: [Graph Figures in matplotlib.ipynb](#)

### Output:

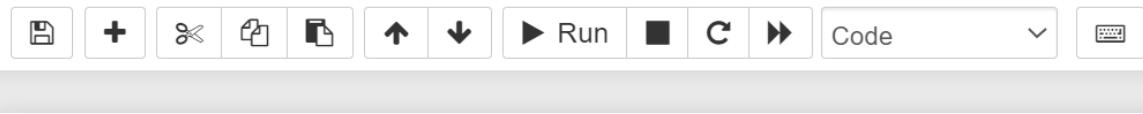


## Subplots in matplotlib

The **Matplotlib subplot()** function can be called to plot two or more plots in one figure. Matplotlib supports all kind of subplots including 2x1 vertical, 2x1 horizontal or a 2x2 grid.

The subplot() function takes three arguments that describes the layout of the figure. The layout is organized in rows and columns, which are represented by the *first* and *second* argument. The third argument represents the index of the current plot.

```
plt.subplot(1, 2, 1)  
#the figure has 1 row, 2 columns, and this plot is the first plot.  
  
plt.subplot(1, 2, 2)  
#the figure has 1 row, 2 columns, and this plot is the second plot.
```



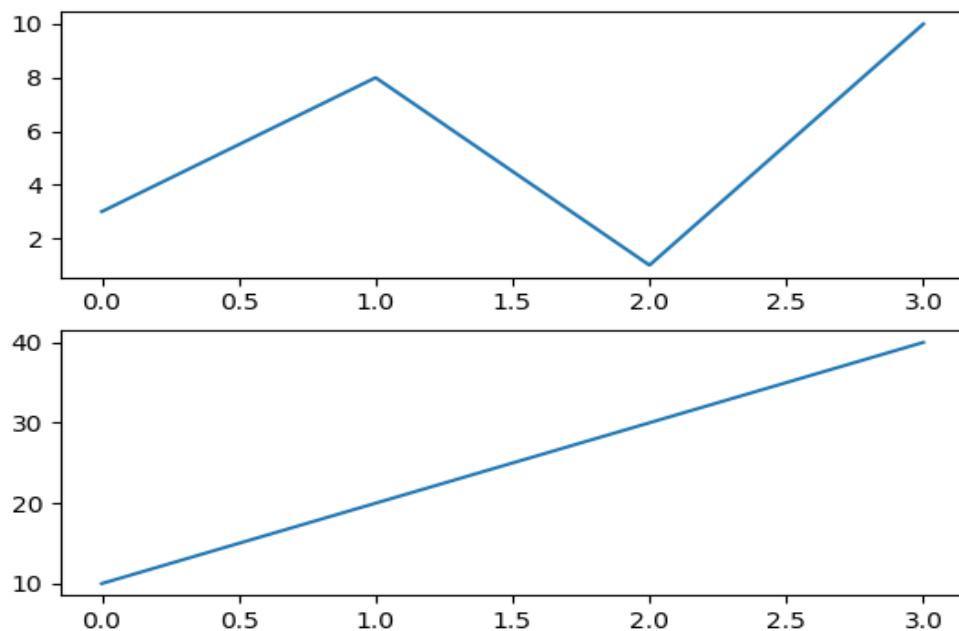
```
In [1]: import matplotlib.pyplot as plt
import numpy as np
#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])
plt.subplot(2, 1, 1)
plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 1, 2)
plt.plot(x,y)
```

Code: [subplot theory.ipynb](#)

## Output



You can draw as many plots you like on one figure, just describe the number of rows, columns, and the index of the plot.

## 3.13 Colors, Markers and line styles

### Matplotlib Markers

You can use the keyword argument marker to emphasize each point with a specified marker:

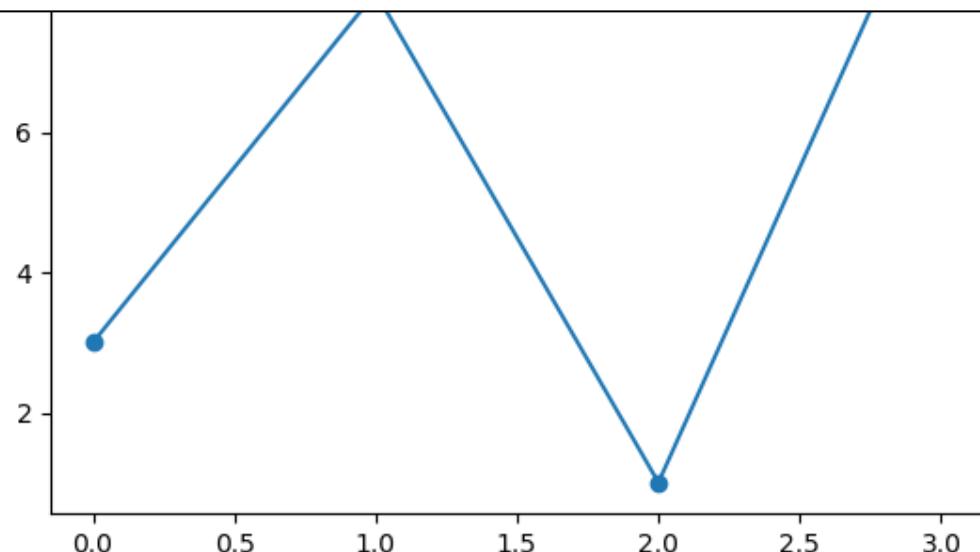
Example:

Mark each point with a circle

```
import matplotlib.pyplot as plt
import numpy as np

y whole points = np.array([3, 8, 1, 10])

plt.plot(whole points, marker = 'o')
plt.show()
```



### Matplotlib Line:

**Linestyle** is keyword used to change the style of the plotted line:

Shorter form of `linestyle` is `ls`.

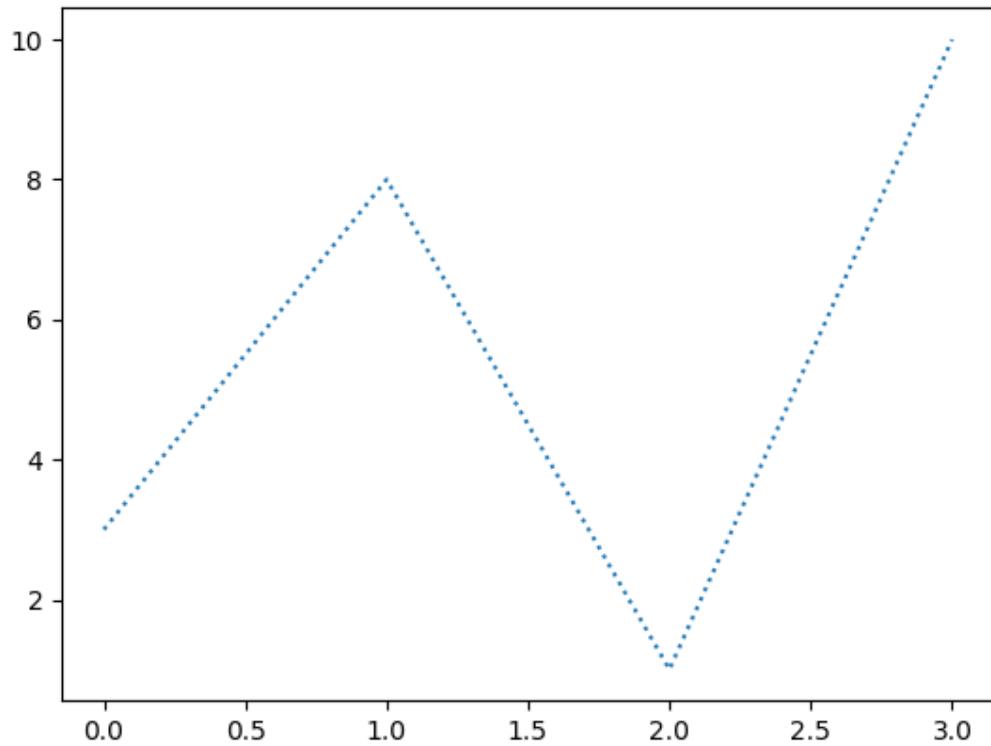
Example:

use dotted line for plot the graph.

```
import matplotlib.pyplot as plt
import numpy as np

y whole points = np.array([3, 8, 1, 10])

plt.plot(whole points, linestyle = 'dotted')
plt.show()
```



### Line Color:

You can use the keyword argument `color` or the shorter `c` to set the color of the line:

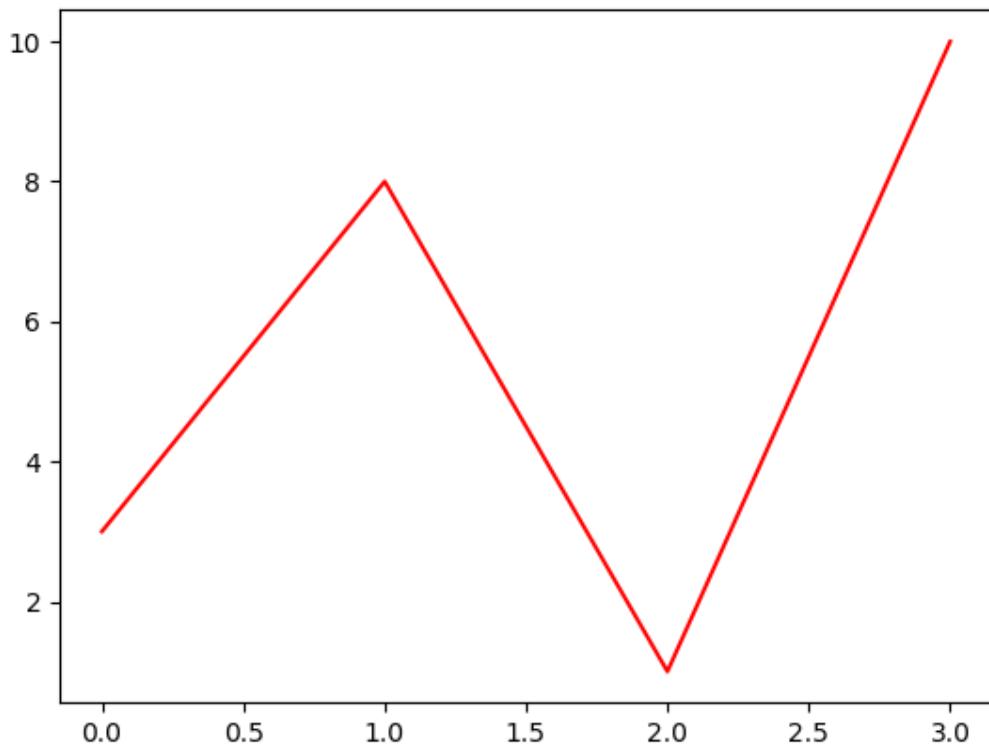
Example:

set the line color red:

```
import matplotlib.pyplot as plt
import numpy as np

y whole points = np.array([3, 8, 1, 10])

plt.plot(y whole points, color = 'r')
plt.show()
```



## Marker Color

You can use the keyword argument `markeredgecolor` or the shorter `mec` to set the color of the edge of the markers:

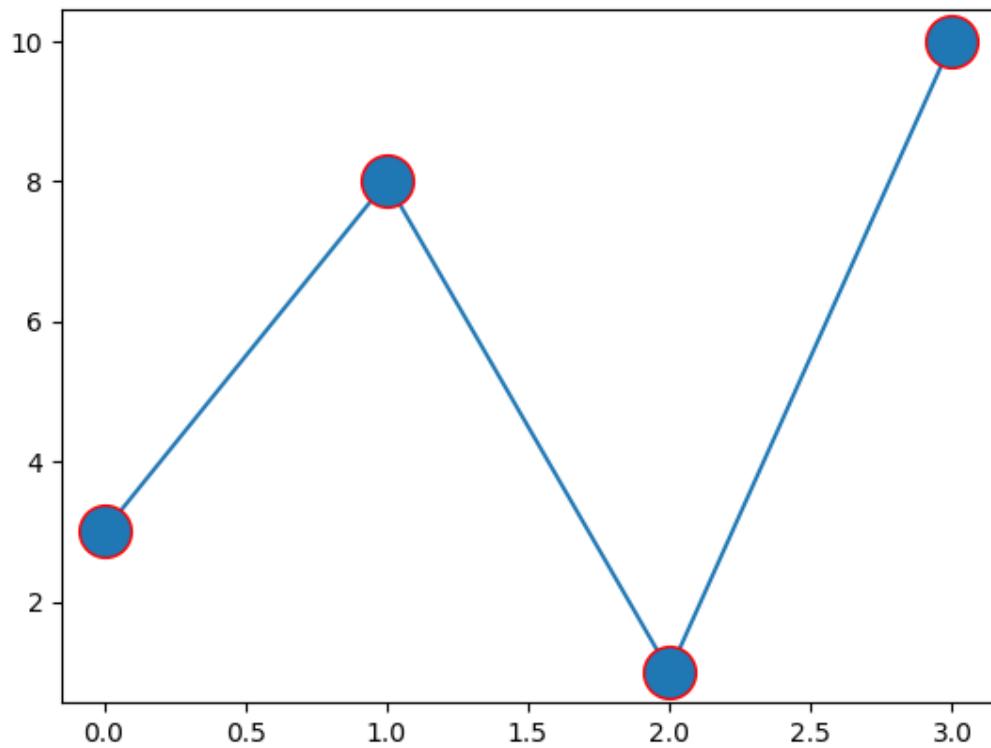
Example:

set the EDGE color to red:

```
import matplotlib.pyplot as plt
import numpy as np

y whole points = np.array([3, 8, 1, 10])

plt.plot(y whole points, marker = 'o', ms = 20, mec = 'r')
plt.show()
```



### 3.14 Matplotlib Configuration

#### Installation process of matplotlib

Install Matplotlib with pip

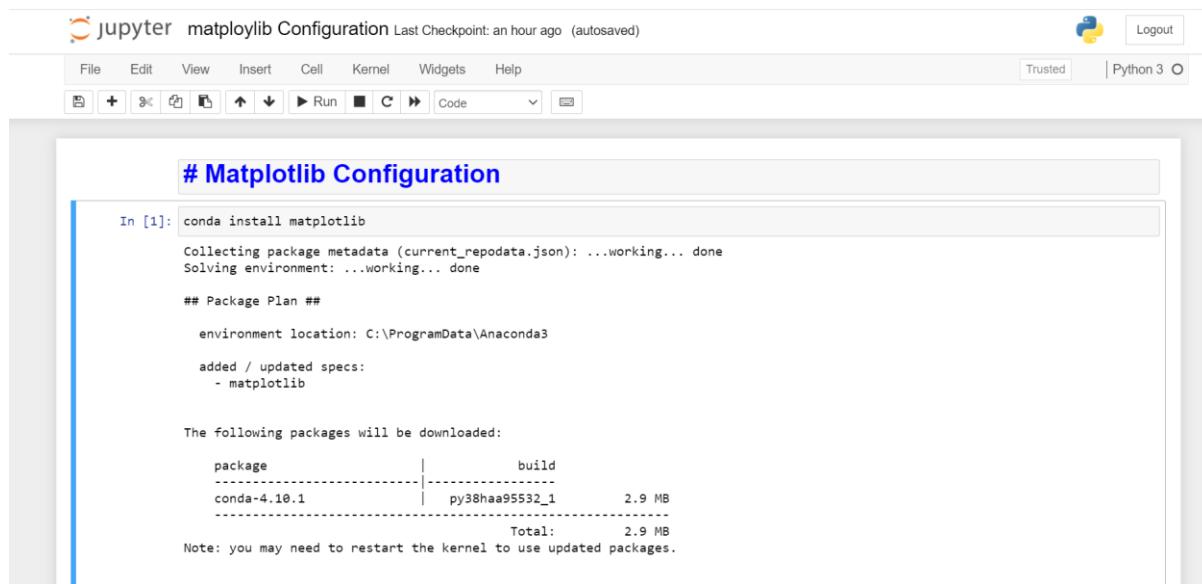
Matplotlib can also be installed using the Python package manager, pip. To install Matplotlib with pip, open a terminal window and type:

```
pip install matplotlib
```

Install Matplotlib with the **Anaconda Prompt**

Matplotlib can be installed using with the Anaconda Prompt. If the Anaconda Prompt is available on your machine, it can usually be seen in the Windows Start Menu. To install Matplotlib, open the Anaconda Prompt and type:

```
conda install matplotlib
```



The screenshot shows a Jupyter Notebook interface. The title bar says "jupyter matplotlib Configuration Last Checkpoint: an hour ago (autosaved)". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and various icons for file operations. A status bar at the bottom right shows "Trusted" and "Python 3". The main area has a title "# Matplotlib Configuration". In cell [1], the code "conda install matplotlib" is run, showing the package being collected and solved. It then displays the package plan, showing "matplotlib" added. It lists packages to be downloaded, including "conda-4.10.1" and "py38haa95532\_1". A note at the bottom says "Note: you may need to restart the kernel to use updated packages."

After the installation is completed. Let's start using Matplotlib with **Jupyter Notebook**.

We will be plotting various graphs in the **Jupyter Notebook** using **Matplotlib**.

## 3.15 Line Plot, Bar Plot, Scatter Plot

### Line Plot

Write below code in **jupyter notebook** for **Line chart** and click on **Run**

```
#Line plot using matplotlib librabry, here we use pyplot function of matplotlib
# importing matplotlib module
from matplotlib import pyplot as plt

# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

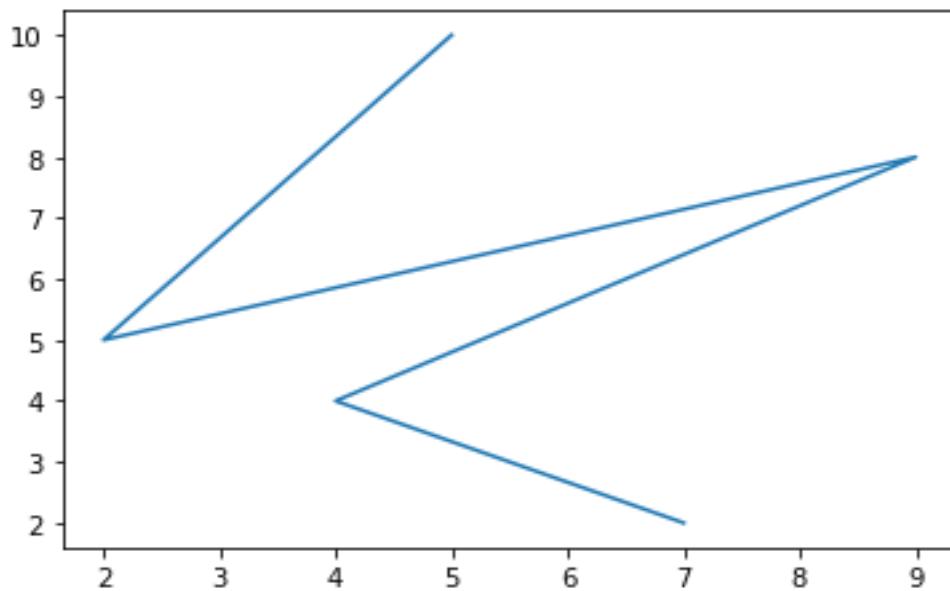
# Function to plot
plt.plot(x, y)

# function to show the plot
plt.show()
```

Code: [Line Plot.ipynb](#)

Result:

When you run the above code in jupyter notebook you will get the output shown below.



## Bar Plot

Write the code given below in **jupyter notebook** for **Bar chart** and click on **Run**



```
In [ ]: # Bar Plot

# importing matplotlib module
from matplotlib import pyplot as plt

# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

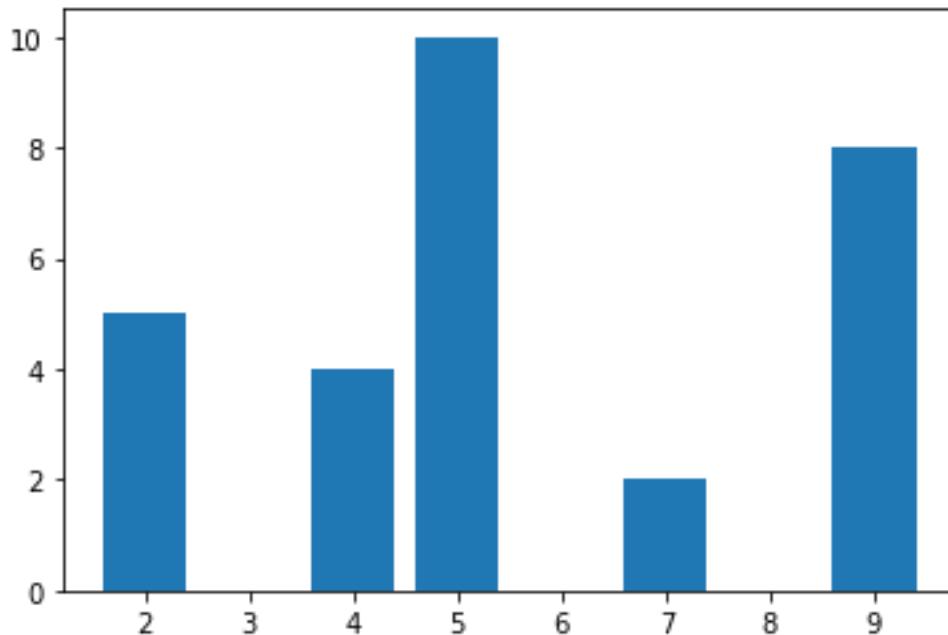
# Function to plot
plt.bar(x, y)

# function to show the plot
plt.show()
```

Code: [Bar Plot.ipynb](#)

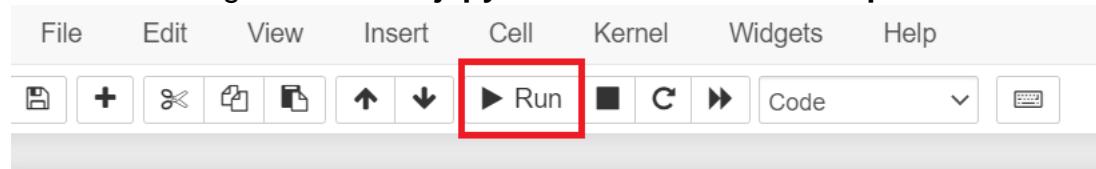
Output:

When you run the above code in jupyter notebook you will get the output given below.



## Scatter Plot

Write the code given below in **jupyter notebook** for **Scatter plot** and click on **Run**



```
In [2]: #Scatter Plot
# importing matplotlib module
from matplotlib import pyplot as plt

# x-axis values
x = [5, 2, 9, 4, 7, 3, 8, 10, 1]

# Y-axis values
y = [10, 5, 8, 4, 2, 1, 6, 3, 9]

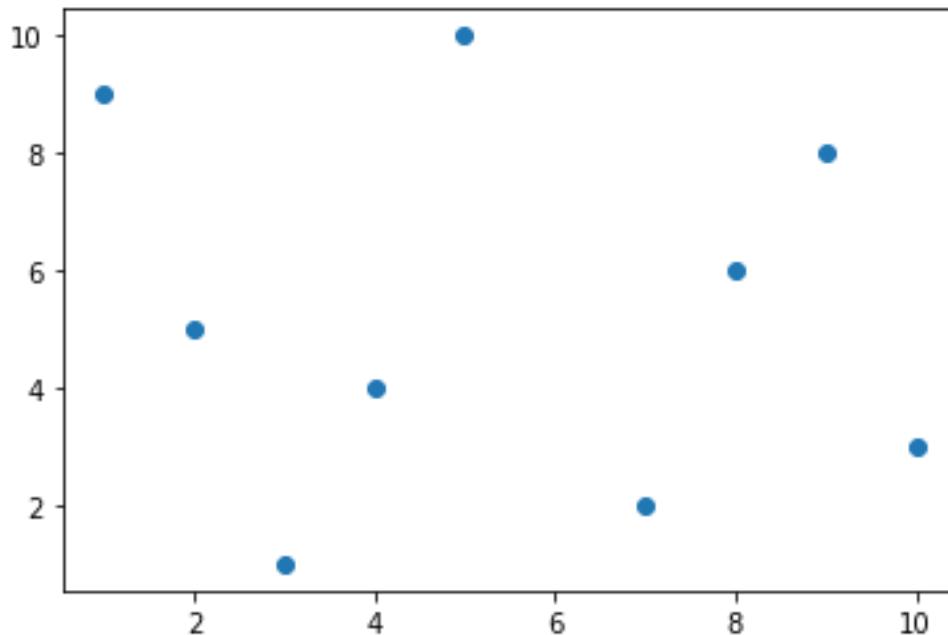
# Function to plot scatter
plt.scatter(x, y)

# function to show the plot
plt.show()
```

Code: [Scatter Plot.ipynb](#)

Result:

When you run the above code in jupyter notebook you will get the following output.



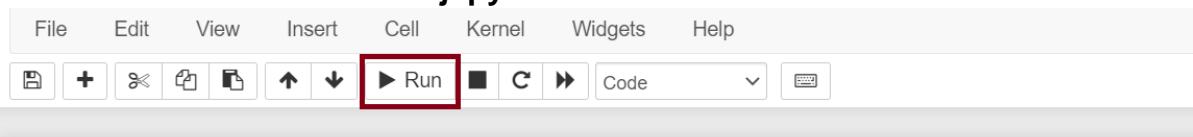
## 3.16 Ticks, Labels and Legends, subplots

### Ticks and Labels

**Ticks** are the values used to show specific points on the coordinate axis. It can be a number or a string. Whenever we plot a graph, the axes adjust and take the default ticks. Matplotlib's default ticks are generally sufficient in common situations but are in no way optimal for every plot.

Here, we will see how to customize these ticks as per our need.

Write the code shown below in **jupyter notebook** and click on **Run**.



```
In [2]: # Ticks are the markers denoting data points on axes.
# importing libraries
import matplotlib.pyplot as plt
import numpy as np

# values of x and y axes
x = [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
y = [1, 4, 3, 2, 7, 6, 9, 8, 10, 5]

plt.plot(x, y, 'b')
plt.xlabel('x')
plt.ylabel('y')
# here we set the size for ticks, rotation and color value

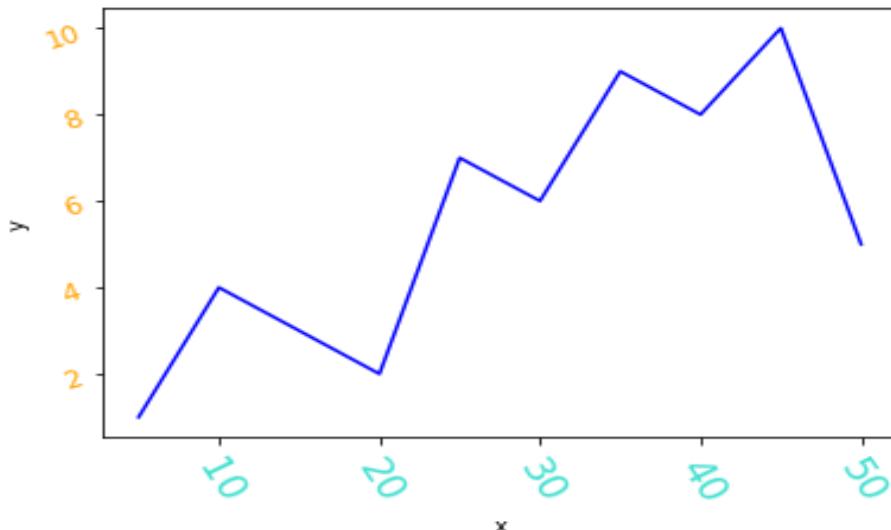
plt.tick_params(axis="x", labelsize=18, labelrotation=-60, labelcolor="turquoise")
plt.tick_params(axis="y", labelsize=12, labelrotation=20, labelcolor="orange")

plt.show()
```

Code: [Ticks.ipynb](#)

Output:

When you run the above code in jupyter notebook you will get the following output.



### Adding title and Labelling the Axes in the graph

Write the code given below in **jupyter notebook** to add title and label to the graph and click on **Run**.



```
In [1]: #Adding title and Labelling the Axes in the graph
```

```
# importing matplotlib module
from matplotlib import pyplot as plt

# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot
plt.bar(x, y)
```

Title

```
# Adding Title
plt.title("Bar graph ")
```

Labeling

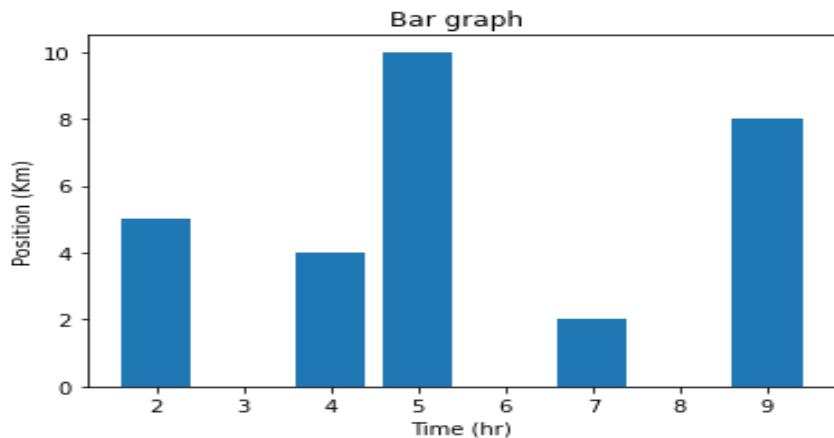
```
# Labeling the axes
plt.xlabel("Time (hr)")
plt.ylabel("Position (Km)")
```

```
# function to show the plot
plt.show()
```

Code: [Labelling.ipynb](#)

Result:

When you run the above code in jupyter notebook you will get the following result.



## Adding Legend in the graph

A legend is an area describing the elements of the graph. In the matplotlib library, there's a function called **legend()** which is used to place a legend on the axes.

### Matplotlib.pyplot.legend()

Write the code given below in **jupyter notebook** to add Legend to the graph and click on **Run**.

File Edit View Insert Cell Kernel Widgets Help

Run

```
In [1]: #Adding Legend in the graph
# importing modules
import numpy as np
import matplotlib.pyplot as plt

# Y-axis values
y1 = [2, 3, 4.5]

# Y-axis values
y2 = [1, 1.5, 5]

# Function to plot
plt.plot(y1)
plt.plot(y2)

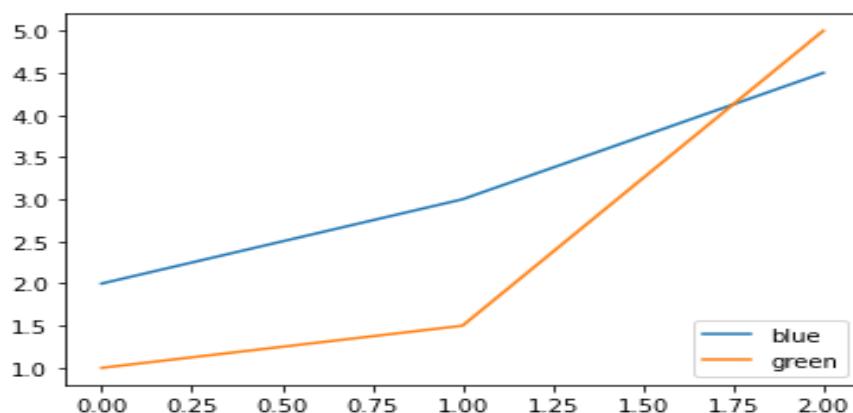
# Function add a legend
plt.legend(["blue", "green"], loc ="lower right")

# function to show the plot
plt.show()
```

Legend

Code: [Legends.ipynb](#)

Result:



## Display Multiple Plots using subplot() function

With the **subplots()** function you can draw multiple plots in one figure

File Edit View Insert Cell Kernel Widgets Help

Save and Checkpoint

In [1]:

```
#subplots
import matplotlib.pyplot as plt
import numpy as np

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
plt.plot(x,y)

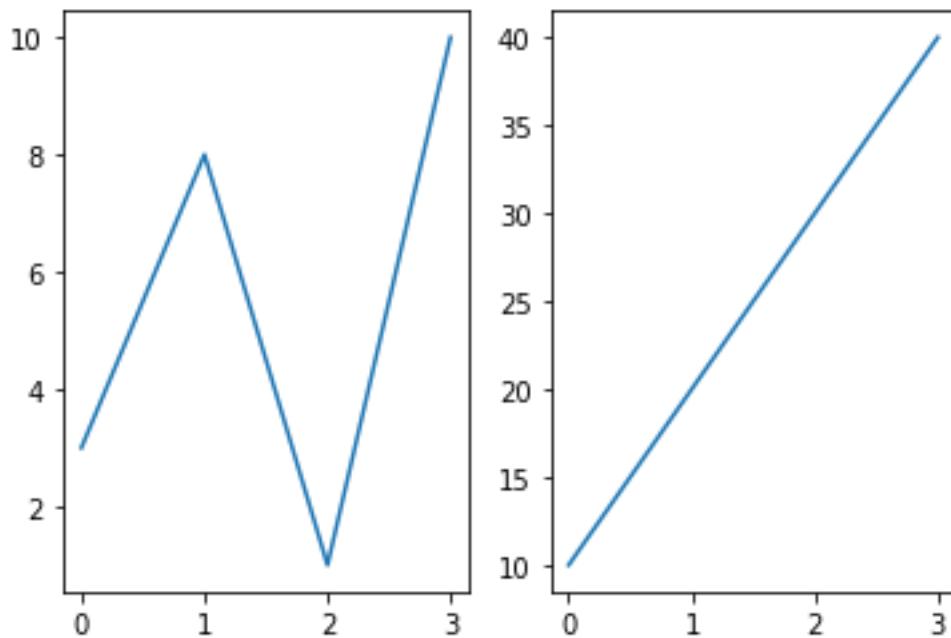
#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)

plt.show()
```

Code: subplot.ipynb

Result:



## 3.17 Histograms and Binning

A histogram is an accurate graphical representation of the distribution of numerical data. It is an estimate of the probability distribution of a continuous variable (quantitative variable) and was first introduced by Karl Pearson. It is a kind of bar graph.

To construct a histogram, the first step is to “bin” the range of values — that is, divide the entire range of values into a series of intervals — and then count how many values fall into each interval. The bins are usually specified as consecutive, non-overlapping intervals of a variable. The bins (intervals) must be adjacent, and are often (but are not required to be) of equal size.

Bins are also sometimes called "intervals", "classes", or "buckets".

The heights of the bars tell us how many data points are in each bin.

For example, this histogram says that there are 8 pumpkins whose mass is between 6 and 9 kilograms.

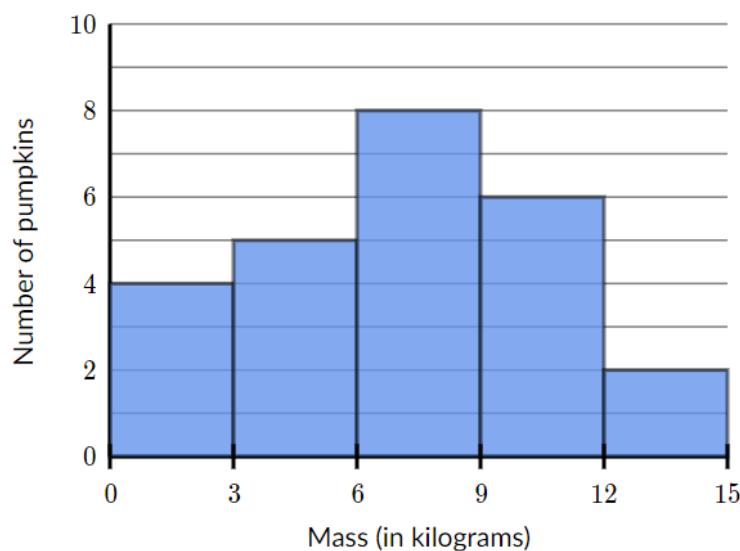
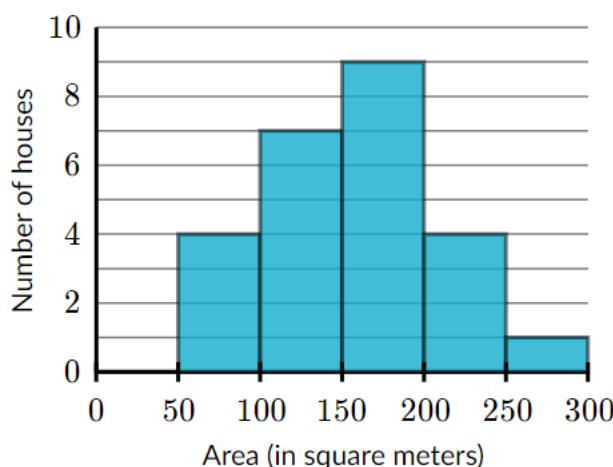


Image: Histogram

Reference:<https://www.khanacademy.org/math/statistics-probability/displaying-describing-data/quantitative-data-graphs/a/histograms-review>

Size of each house on Prince Street



### Exercise:

Observe the histogram and find out how many Houses are there in Prince Street?

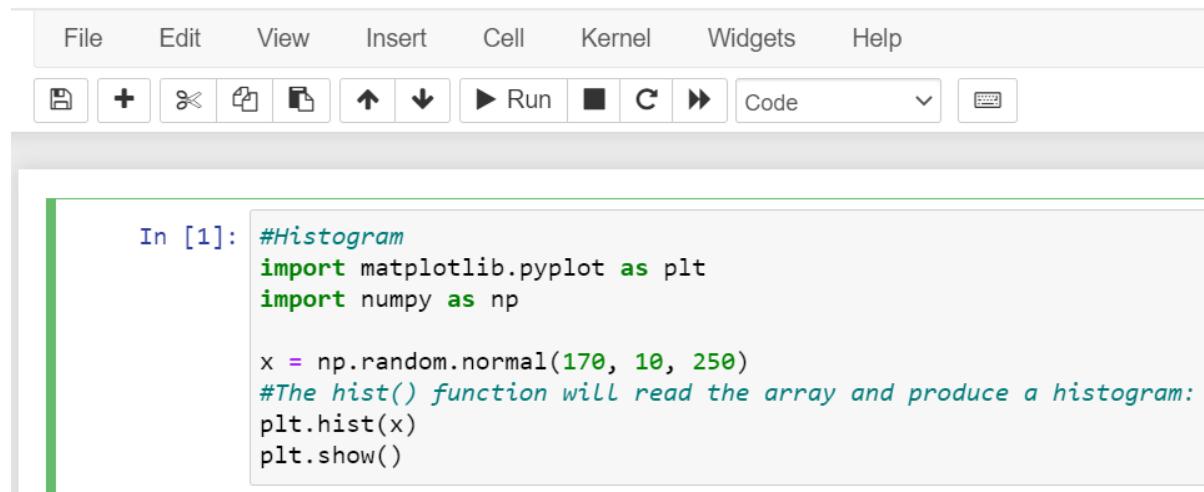
Answer: \_\_\_\_\_

## Create Histogram in Python using matplotlib

In Matplotlib, we use the `hist()` function to create histograms.

The `hist()` function will use an array of numbers to create a histogram, the array is sent into the function as an argument.

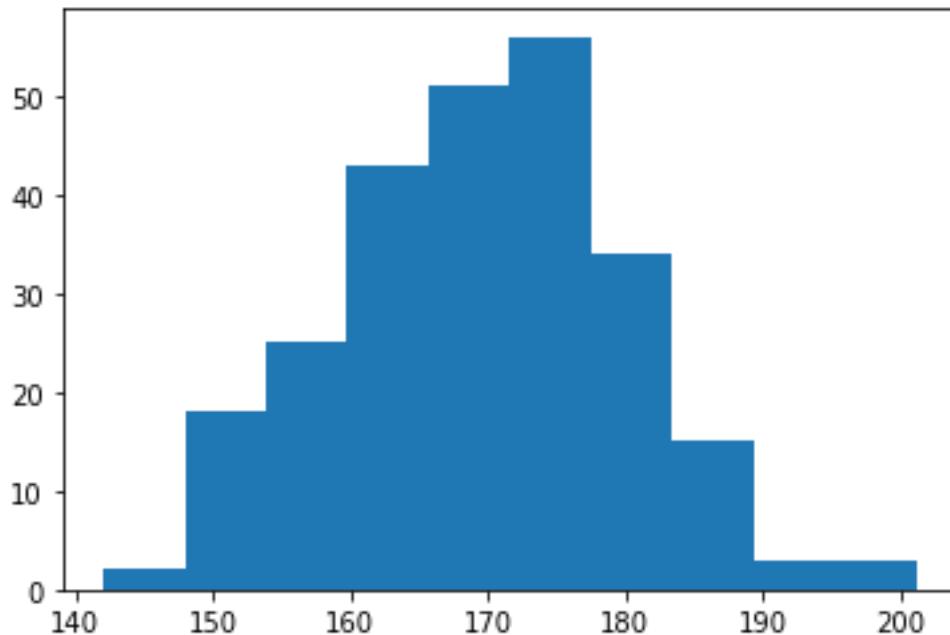
The `hist()` function will read the array and produce a histogram:



```
In [1]: #Histogram
import matplotlib.pyplot as plt
import numpy as np

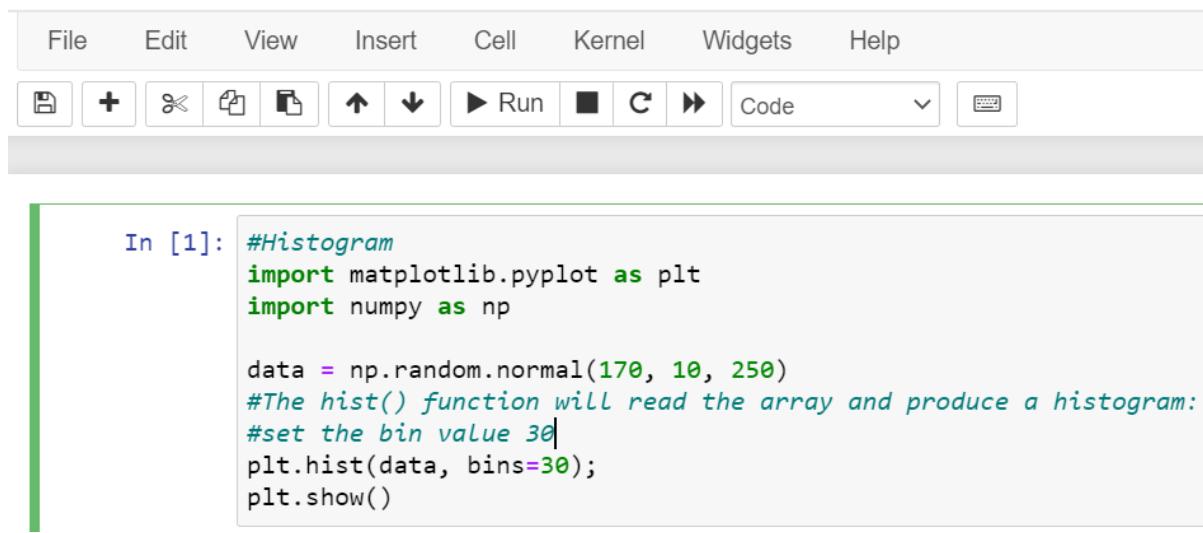
x = np.random.normal(170, 10, 250)
#The hist() function will read the array and produce a histogram:
plt.hist(x)
plt.show()
```

Output:



## Binning

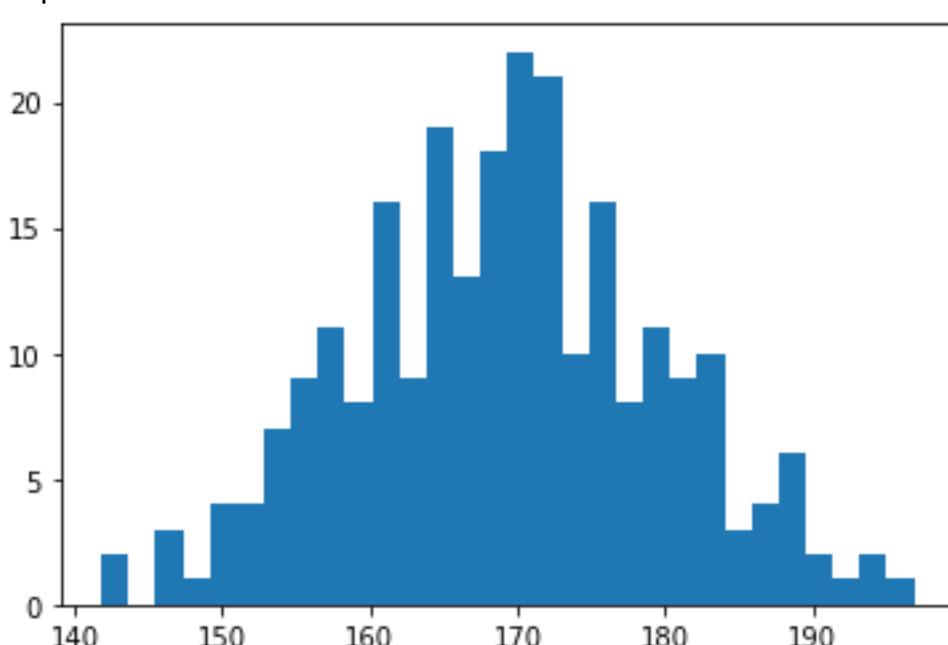
Set the `bins` value to the above histogram



In [1]: `#Histogram  
import matplotlib.pyplot as plt  
import numpy as np  
  
data = np.random.normal(170, 10, 250)  
#The hist() function will read the array and produce a histogram:  
#set the bin value 30  
plt.hist(data, bins=30);  
plt.show()`

Code: [hist\\_bin.ipynb](#)

Output:



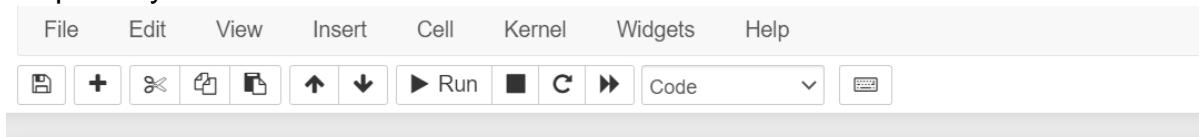
### 3.18 Text and Annotations

Create an annotation: a piece of text referring to a data point.

Creating a good visualization involves guiding the reader so that the figure tells a story. In some cases, this story can be told in an entirely visual manner, without the need for added text, but in others, small textual cues and labels are necessary. Perhaps the most basic types of annotations you will use are axes labels and titles, but the options go beyond this. Let's take a look at some data and how we might visualize and annotate it to help convey interesting information.

Let's try to understand it by one example.

The **annotate()** function in pyplot module of matplotlib library is used to annotate the point  $xy$  with texts



```
In [2]: #import matplotlib.pyplot
import matplotlib.pyplot as plt
import numpy as np

#draw plot as fig
fig, ax = plt.subplots()

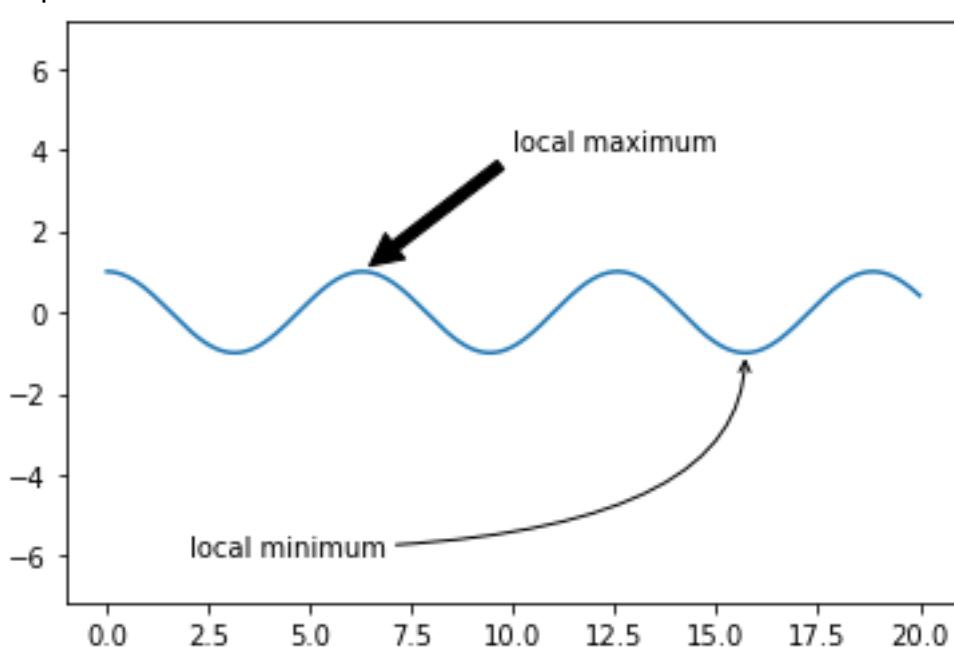
x = np.linspace(0, 20, 1000)
ax.plot(x, np.cos(x))
ax.axis('equal')
#text annotation to the plot where it indicate maximum value of the curve
ax.annotate('local maximum', xy=(6.28, 1), xytext=(10, 4),
           arrowprops=dict(facecolor='black', shrink=0.05))
#text annotation to the plot where it indicate minimum value of the curve
ax.annotate('local minimum', xy=(5 * np.pi, -1), xytext=(2, -6),
           arrowprops=dict(arrowstyle="->",
                          connectionstyle="angle3,angleA=0,angleB=-90"));


```

Code: [text and annotate.ipynb](#)

Here annotation mark is the simple arrow.

Output



In above example you can observe text annotate local maximum and local minimum by arrow

Example 2

File Edit View Insert Cell Kernel Widgets Help

File New Open Save Run Cell Kernel Widgets Help

In [3]:

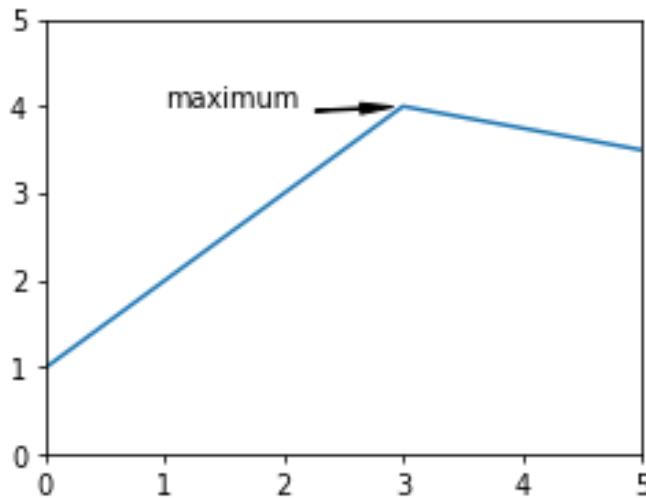
```
import matplotlib.pyplot as plt
w = 4
h = 3
d = 70
plt.figure(figsize=(w, h), dpi=d)
plt.axis([0, 5, 0, 5])
x = [0, 3, 5]
y = [1, 4, 3.5]
label_x = 1
label_y = 4
arrow_x = 3
arrow_y = 4

arrow_properties = dict(
    facecolor="black", width=0.5,
    headwidth=4, shrink=0.1)

plt.annotate(
    "maximum", xy=(arrow_x, arrow_y),
    xytext=(label_x, label_y),
    arrowprops=arrow_properties)
plt.plot(x, y)
```

Code: [text and annotate.ipynb](#)

Output:



### 3.19 Three-Dimensional Plotting in Matplotlib

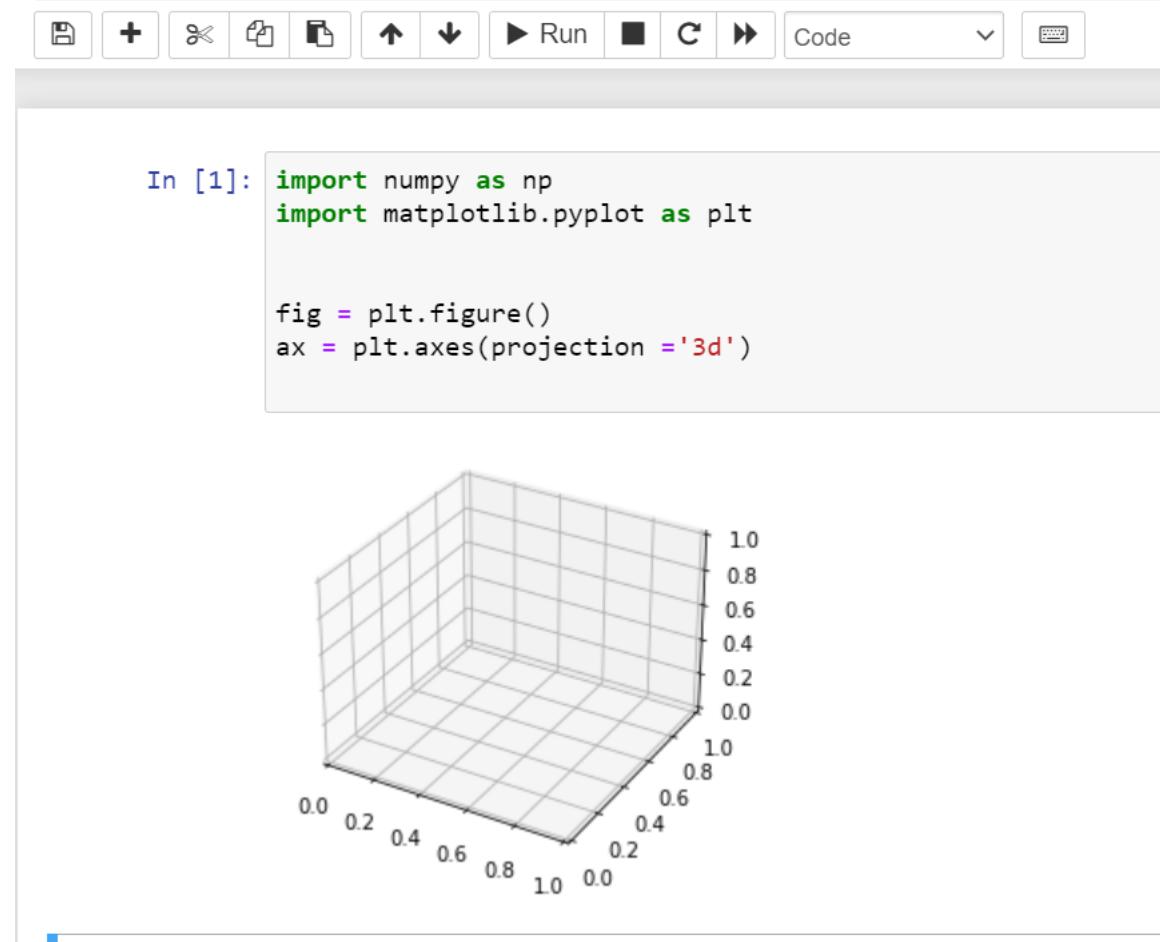
Three-dimensional plots are enabled by importing the mplot3d toolkit, included with the main Matplotlib installation:

Step 1:

```
from mpl_toolkits import mplot3d
```

Once this submodule is imported, three-dimensional axes can be created by passing the keyword `projection='3d'` to any of the normal axes creation routines:

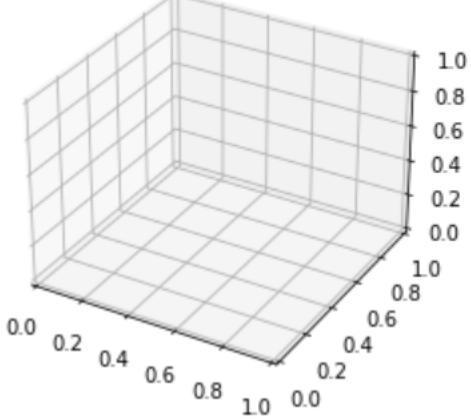
Step 2: create simple 3D graph



In [1]:

```
import numpy as np
import matplotlib.pyplot as plt

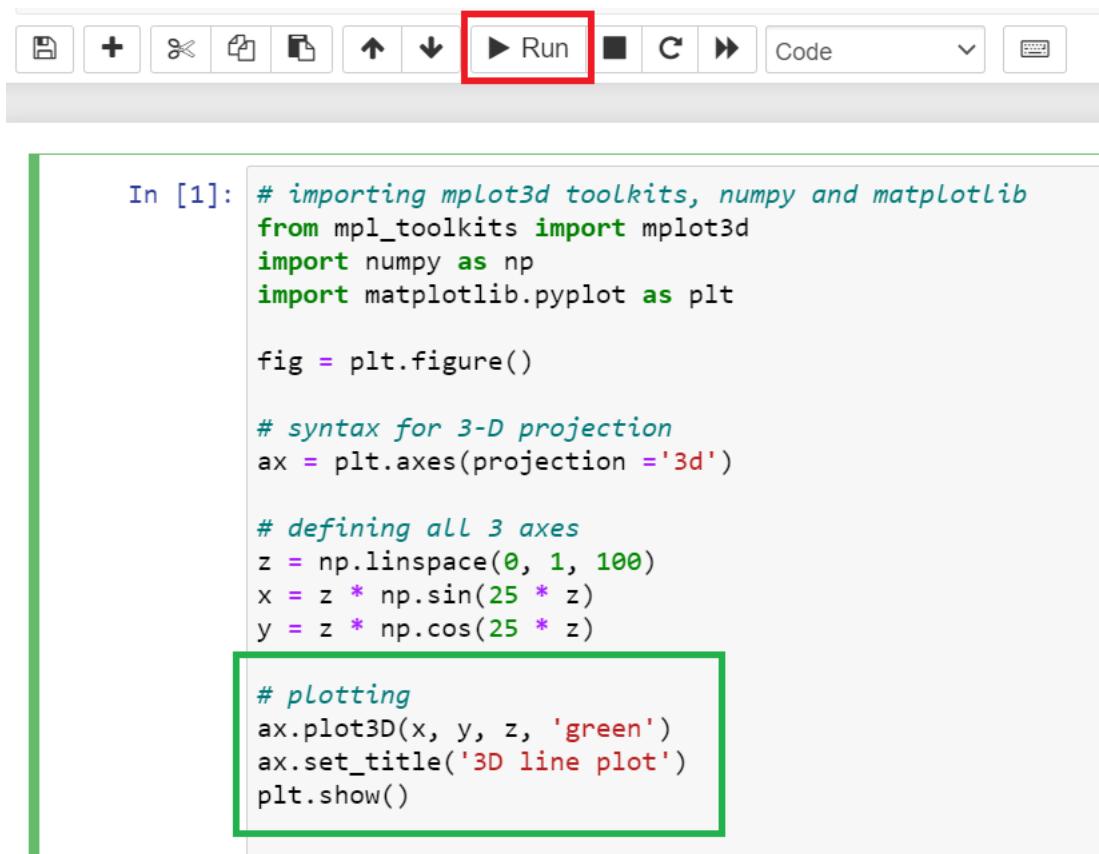
fig = plt.figure()
ax = plt.axes(projection ='3d')
```



Code: [3D plotting matplotlib.ipynb](#)

## Plotting 3-D Lines and Points

**Graph with lines and point** are the simplest 3 dimensional graph. `ax.plot3d` and `ax.scatter` are the function to plot line and point graph respectively.  
Write down the code given below in jupyter notebook and click on Run



```
In [1]: # importing matplotlib toolkits, numpy and matplotlib
from mpl_toolkits import mplot3d
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure()

# syntax for 3-D projection
ax = plt.axes(projection ='3d')

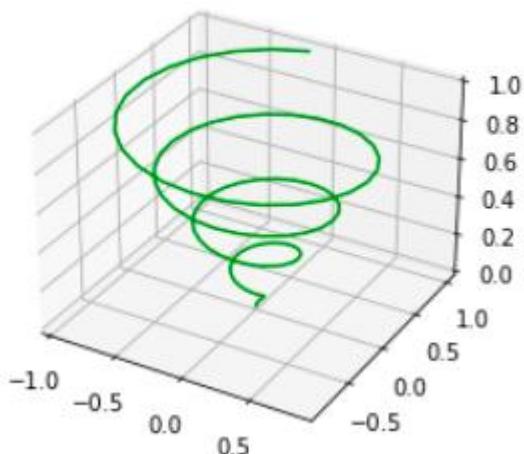
# defining all 3 axes
z = np.linspace(0, 1, 100)
x = z * np.sin(25 * z)
y = z * np.cos(25 * z)

# plotting
ax.plot3D(x, y, z, 'green')
ax.set_title('3D line plot')
plt.show()
```

Code: [Plotting 3-D Lines and Points.ipynb](#)

Output:

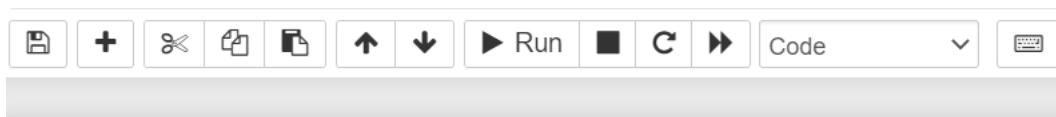
3D line plot



## Plot a pie chart in Python using Matplotlib

Matplotlib API has `pie()` function in its `pyplot` module which creates a pie chart representing the data in an array.

Let's create a simple pie chart using the `pie()` function:



```
In [1]: # Import Libraries
from matplotlib import pyplot as plt
import numpy as np

# Creating dataset
cars = ['AUDI', 'BMW', 'FORD',
        'TESLA', 'JAGUAR', 'MERCEDES']

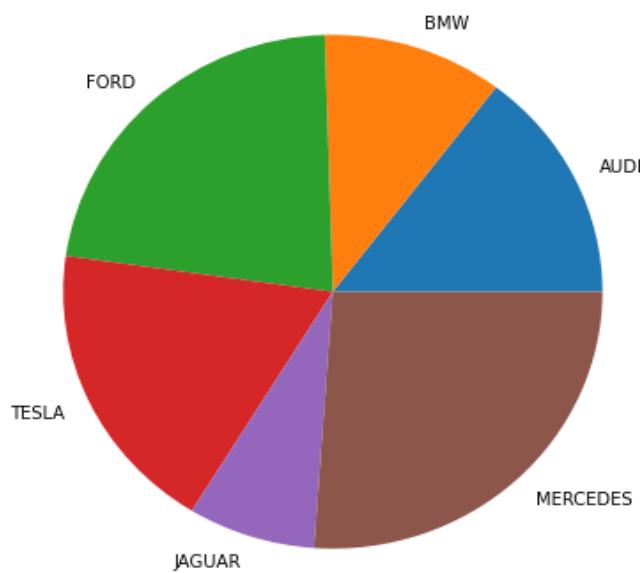
data = [23, 17, 35, 29, 12, 41]

# Creating plot
fig = plt.figure(figsize =(10, 7))
plt.pie(data, labels = cars)

# show plot
plt.show()
```

Code: [pie chart.ipynb](#)

Output:

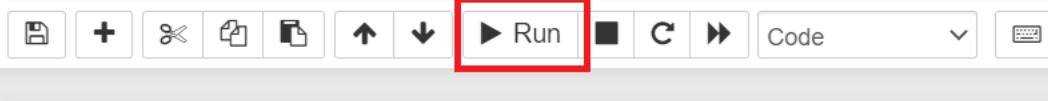


## 3.20 Visualizing with NumPy

### Create plots using NumPy Array

NumPy is a Python library used for working with arrays.

Step 1: write the program given below in jupyter notebook and click on Run.

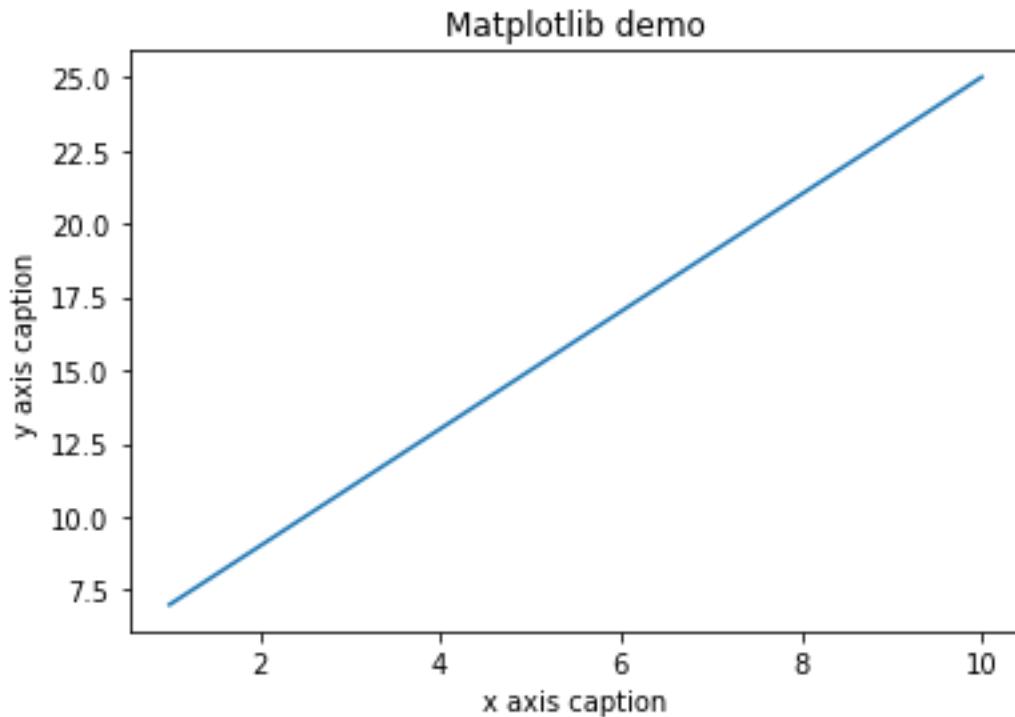


```
In [1]: #import numpy
         import numpy as np
         from matplotlib import pyplot as plt

         x = np.arange(1,11)
         y = 2 * x + 5
         plt.title("Matplotlib demo")
         plt.xlabel("x axis caption")
         plt.ylabel("y axis caption")
         plt.plot(x,y)
         plt.show()
```

Code: [Numpy excercise.ipynb](#)

Output:



## Draw Sine Wave Plot using Numpy

Write down the code given below in Jupyter notebook and click on Run

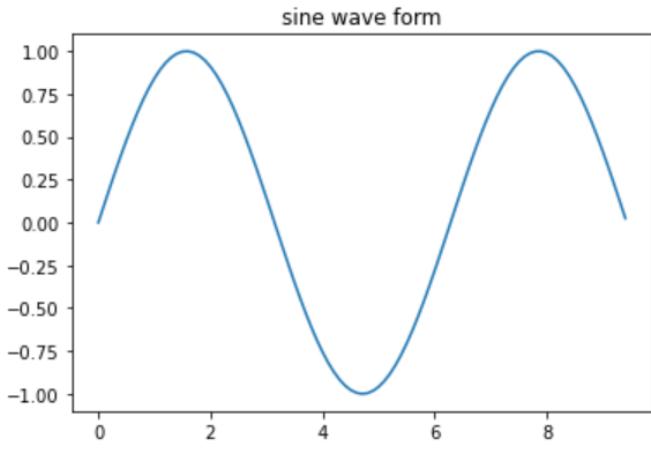
File Edit View Insert Cell Kernel Widgets Help

File + × ↻ ↺ ⌛ Run Cell Code

```
In [1]: import numpy as np
import matplotlib.pyplot as plt

# Compute the x and y coordinates for points on a sine curve
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)
plt.title("sine wave form")

# Plot the points using matplotlib
plt.plot(x, y)
plt.show()
```



Code: [sine\\_wave\\_numpy.ipynb](#)

We have learned and explored the various functions in Matplotlib library which are used to visualize the data in various format and make it available to help in decision making process. Now, let us see another powerful library called Pandas which is very popular in data analytics.

### 3.21 Data Manipulation with Pandas

**Pandas** is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool. Pandas is a newer package built on top of NumPy, and provides an efficient implementation of a **DataFrame**.

**DataFrames** are essentially multidimensional arrays with attached row and column labels, and often with heterogeneous types and/or missing data. Apart from offering a convenient storage interface for labelled data, Pandas also implements a number of powerful data operations familiar to users of both database frameworks and spreadsheet programs.

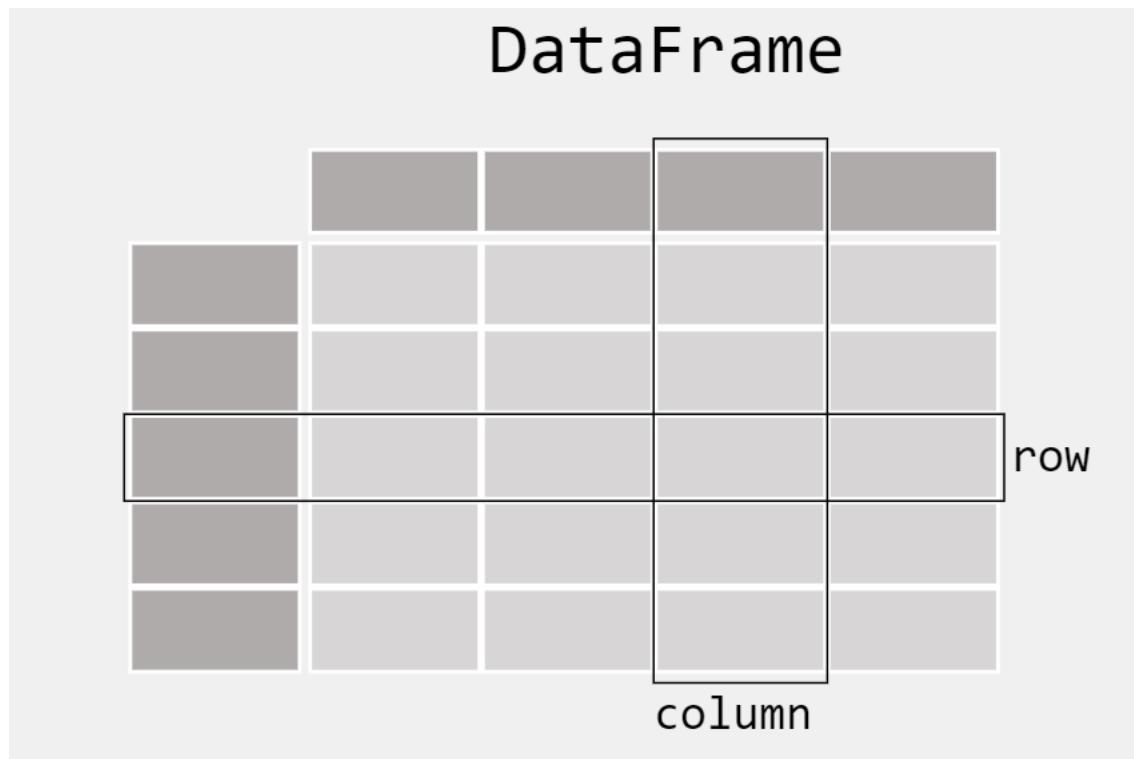


Image: dataframe

Reference: [https://pandas.pydata.org/docs/getting\\_started/index.html](https://pandas.pydata.org/docs/getting_started/index.html)

## How to install pandas

**import pandas as pd**

To load the pandas package and start working with it, import the package

For e.g store passengers data of the Titanic. For a number of passengers, I know the name (characters), age (integers) and sex (male/female) data.

```
In [7]: import pandas as pd
df = pd.DataFrame(
    {
        "Name": [
            "Mr. Pravinkumar Prajapati",
            "Miss. Priyanka Paul",
            "Miss. Elizabeth",
        ],
        "Age": [22, 30, 58],
        "Sex": ["male", "female", "female"],
    }
)
```

Code: [panda data manipulation.ipynb](#)

Output:

In [8]: df

Out[8]:

	Name	Age	Sex
0	Mr. Pravinkumar Prajapati	22	male
1	Miss. Priyanka Paul	30	female
2	Miss. Elizabeth	58	female

To manually store data in a table, create a [DataFrame](#). When using a Python dictionary of lists, the dictionary keys will be used as column headers and the values in each list as columns of the [DataFrame](#).

Now we can perform more operation to get more insights into data. Let us explore more command

When selecting a single column of a pandas [DataFrame](#), the result is a pandas [Series](#). To select the column, use the column label in between square brackets [].

In [6]: df["Age"]

Out[6]: 0 22  
1 35  
2 58  
Name: Age, dtype: int64

Find maximum Age of the passengers.

In [9]: df["Age"].max()

Out[9]: 58

The [describe\(\)](#) method provides a quick overview of the numerical data in a DataFrame. As the Name and Sex columns are textual data, these are by default not taken into account by the [describe\(\)](#) method

```
In [10]: df.describe()
```

```
Out[10]:
```

Age	
count	3.000000
mean	36.666667
std	18.903263
min	22.000000
25%	26.000000
50%	30.000000
75%	44.000000
max	58.000000

## How to manipulate textual data?

Here we use the Titanic data set, stored as CSV. The data consists of the following data columns:

- **PassengerId**: Id of every passenger.
- **Survived**: These features have value 0 and 1. 0 for not survived and 1 for survived.
- **Pclass**: There are 3 classes: Class 1, Class 2 and Class 3.
- **Name**: Name of passenger.
- **Sex**: Gender of passenger.
- **Age**: Age of passenger.
- **SibSp**: Indication that passenger have siblings and spouse.
- **Parch**: Whether a passenger is alone or have family.
- **Ticket**: Ticket number of passenger.
- **Fare**: Indicating the fare.
- **Cabin**: The cabin of passenger.
- **Embarked**: The embarked category.

```
In [1]: import pandas as pd
```

```
In [3]: titanic = pd.read_csv("titanic.csv")
```

```
In [4]: titanic.head()
```

```
Out[4]:
```

PassengerId	Survived	Pclass		Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Allen, Mr. William Henry	male	35.0	1	0	113803	53.1000	C123	S
4	5	0	3			35.0	0	0	373450	8.0500	NaN	S

---

Code: [pandas data manipulation.ipynb](#)

Task 1: Make all name characters lowercase.

```
In [5]: titanic["Name"].str.lower()
```

```
Out[5]: 0           braund, mr. owen harris
1       cumings, mrs. john bradley (florence briggs th...
2                           heikkinen, miss. laina
3       futrelle, mrs. jacques heath (lily may peel)
4           allen, mr. william henry
...
886           montvila, rev. juozas
887           graham, miss. margaret edith
888   johnston, miss. catherine helen "carrie"
889           behr, mr. karl howell
890           dooley, mr. patrick
Name: Name, Length: 891, dtype: object
```

Code: [pandas data manipulation.ipynb](#)

**Task 2:** Create a new column Surname that contains the surname of the passengers by extracting the part before the comma.

```
In [6]: titanic["Name"].str.split(",")
```

```
Out[6]: 0           [Braund, Mr. Owen Harris]
1       [Cumings, Mrs. John Bradley (Florence Briggs ...
2           [Heikkinen, Miss. Laina]
3       [Futrelle, Mrs. Jacques Heath (Lily May Peel)]
4           [Allen, Mr. William Henry]
...
886           [Montvila, Rev. Juozas]
887           [Graham, Miss. Margaret Edith]
888   [Johnston, Miss. Catherine Helen "Carrie"]
889           [Behr, Mr. Karl Howell]
890           [Dooley, Mr. Patrick]
Name: Name, Length: 891, dtype: object
```

Code: [pandas data manipulation.ipynb](#)

Using the `series.str.split()` method, each of the values is returned as a list of 2 elements. The first element is the part before the comma and the second element is the part after the comma.

```
In [7]: titanic["Surname"] = titanic["Name"].str.split(",").str.get(0)
```

```
In [8]: titanic["Surname"]
```

```
Out[8]: 0      Braund
        1      Cumings
        2      Heikkinen
        3      Futrelle
        4      Allen
       ...
886    Montvila
887    Graham
888    Johnston
889    Behr
890    Dooley
Name: Surname, Length: 891, dtype: object
```

---

**Code:** [pandas data manipulation.ipynb](#)

By this method and various functions you can perform data manipulation using pandas.

## 3.22 Introducing Pandas Objects

Pandas objects can be thought of as enhanced versions of NumPy structured arrays in which the rows and columns are identified with labels rather than simple integer indices

There are three fundamental Pandas data structures:

1. Series,
2. DataFrame
3. Index.

### The Pandas Series Object

A Pandas Series is a one-dimensional array of indexed data. It can be created from a list or array as follows:

```
In [1]: #Pandas Series Object
import numpy as np
import pandas as pd
data = pd.Series([0.25, 0.5, 0.75, 1.0])
data
```

```
Out[1]: 0    0.25
1    0.50
2    0.75
3    1.00
dtype: float64
```

```
In [ ]:
```

Code: [panda object.ipynb](#)

As we see in the output, the `Series` wraps both a sequence of values and a sequence of indices, which we can access with the `values` and `index` attributes. The `values` are simply a familiar NumPy array:

```
In [2]: data.values
```

```
Out[2]: array([0.25, 0.5 , 0.75, 1. ])
```

Like with a NumPy array, data can be accessed by the associated index via the familiar Python square-bracket notation:

```
In [3]: data[1]
```

```
Out[3]: 0.5
```

```
In [4]: data[1:3]
```

```
Out[4]: 1    0.50
2    0.75
dtype: float64
```

## The Pandas DataFrame Object

The next fundamental structure in Pandas is the `DataFrame`. Like the `Series` object discussed in the previous section, the `DataFrame` can be thought of either as a generalization of a NumPy array, or as a specialization of a Python dictionary.

### *DataFrame as a generalized NumPy array*

If a `Series` is an analog of a one-dimensional array with flexible indices, a `DataFrame` is an analog of a two-dimensional array with both flexible row indices and flexible column names.

Just as you might think of a two-dimensional array as an ordered sequence of aligned one-dimensional columns, you can think of a `DataFrame` as a sequence of aligned `Series` objects. Here, by "aligned" we mean that they share the same index. To demonstrate this, let's first construct two new Series listing of the area and the population of say five states of USA.:

```
In [5]: area_dict = {'California': 423967, 'Texas': 695662, 'New York': 141297,
                  'Florida': 170312, 'Illinois': 149995}
area = pd.Series(area_dict)
area
```

```
Out[5]: California    423967
         Texas       695662
         New York    141297
         Florida     170312
         Illinois    149995
        dtype: int64
```

```
In [7]: population_dict = {'California': 38332521, 'Texas': 26448193, 'New York': 19651127,
                           'Florida': 19552860, 'Illinois': 12882135}
population = pd.Series(population_dict)
population
```

```
Out[7]: California    38332521
         Texas       26448193
         New York    19651127
         Florida     19552860
         Illinois    12882135
        dtype: int64
```

[Code: panda object.ipynb](#)

Now that we have the `area` and the `population` Series from before, we can use a dictionary to construct a single two-dimensional object containing this information:

```
In [8]: states = pd.DataFrame({'population': population,
                               'area': area})
states
```

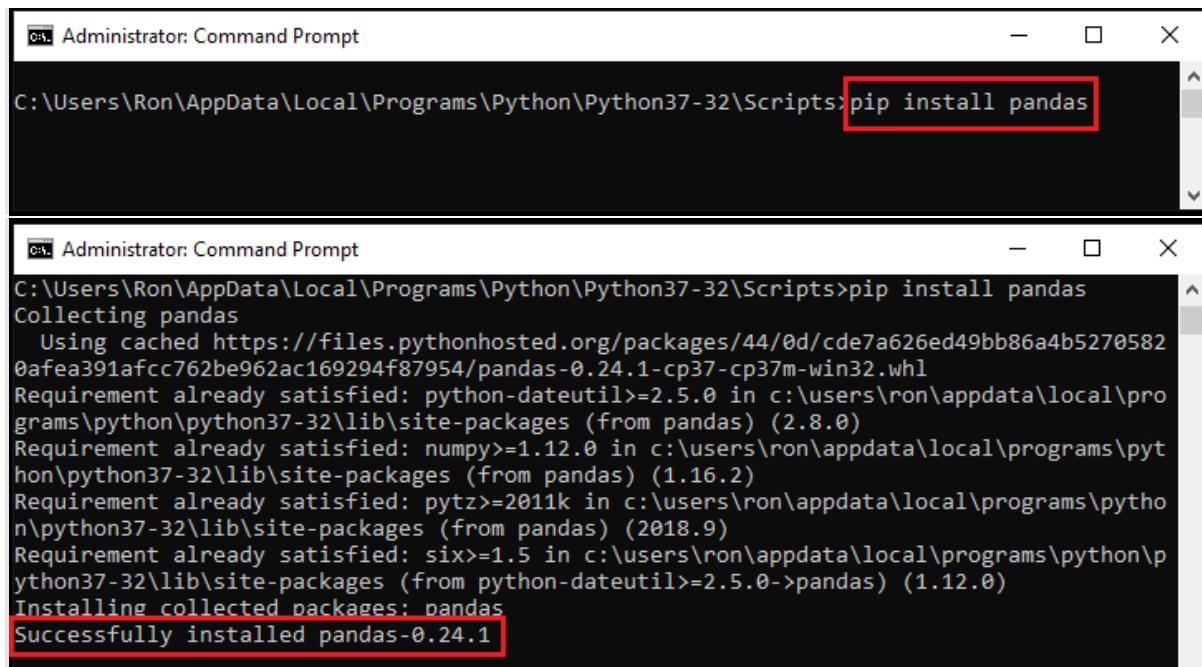
```
Out[8]:
```

	population	area
California	38332521	423967
Texas	26448193	695662
New York	19651127	141297
Florida	19552860	170312
Illinois	12882135	149995

[Code: panda object.ipynb](#)

## Installing Pandas from pip

`pip install pandas`



The screenshot shows two windows of the Windows Command Prompt. Both windows are titled "Administrator: Command Prompt".  
The top window's command line shows the command: C:\Users\Ron\AppData\Local\Programs\Python\Python37-32\Scripts>pip install pandas. This line is highlighted with a red rectangle.  
The bottom window shows the execution of the command:  
Collecting pandas  
Using cached https://files.pythonhosted.org/packages/44/0d/cde7a626ed49bb86a4b52705820afea391afcc762be962ac169294f87954/pandas-0.24.1-cp37-cp37m-win32.whl  
Requirement already satisfied: python-dateutil>=2.5.0 in c:\users\ron\appdata\local\programs\python\python37-32\lib\site-packages (from pandas) (2.8.0)  
Requirement already satisfied: numpy>=1.12.0 in c:\users\ron\appdata\local\programs\python\python37-32\lib\site-packages (from pandas) (1.16.2)  
Requirement already satisfied: pytz>=2011k in c:\users\ron\appdata\local\programs\python\python37-32\lib\site-packages (from pandas) (2018.9)  
Requirement already satisfied: six>=1.5 in c:\users\ron\appdata\local\programs\python\python37-32\lib\site-packages (from python-dateutil>=2.5.0->pandas) (1.12.0)  
Installing collected packages: pandas  
Successfully installed pandas-0.24.1

You can quickly check if the package was successfully installed in Python, by opening the **Python IDLE** and then running the command “import pandas”

If no errors appear, then the package was successfully installed.

For more details on Python packages you can refer <https://packaging.python.org/>

### 3.23 Creating Series from simple datatypes

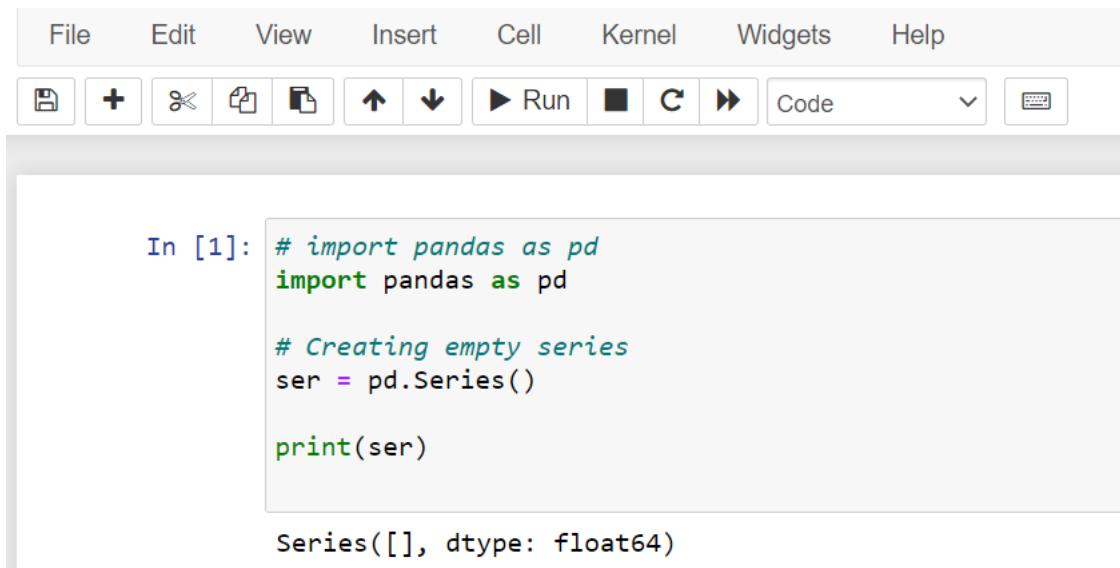
#### Creating a Pandas Series

The Pandas Series can be defined as a one-dimensional array that is capable of storing various data types. We can easily convert the list, tuple, and dictionary into series using "**series**" method. The row labels of series are called the index. A Series cannot contain multiple columns. It has the following parameter:

- **data:** It can be any list, dictionary, or scalar value.
- **index:** The value of the index should be unique and hashable. It must be of the same length as data. If we do not pass any index, default **np.arange(n)** will be used.
- **dtype:** It refers to the data type of series.
- **copy:** It is used for copying the data.

##### 1. Create an Empty Series:

We can easily create an empty series in Pandas which means it will not have any value.



In [1]:

```
# import pandas as pd
import pandas as pd

# Creating empty series
ser = pd.Series()

print(ser)
```

Series([], dtype: float64)

[Code: Creating Series from simple datatypes.ipynb](#)

## 2. Creating a series from array:

In order to create a series from array, we have to import a numpy module and have to use array() function.



In [2]:

```
# import pandas as pd
import pandas as pd

# import numpy as np
import numpy as np

# simple array
data = np.array(['m', 'i', 'c', 'r', 'o', 's', 'o', 'f', 't'])

ser = pd.Series(data)
print(ser)
```

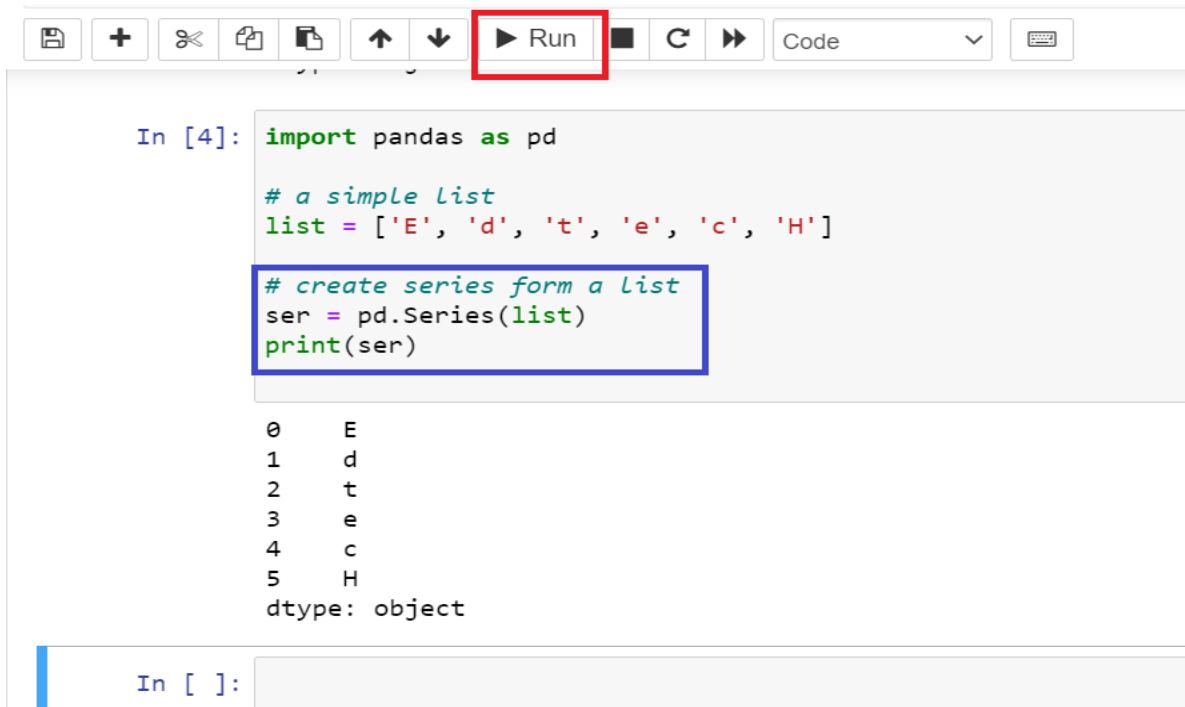
OUTPUT---->

```
0    m
1    i
2    c
3    r
4    o
5    s
6    o
7    f
8    t
dtype: object
```

[Code: Creating Series from simple datatypes.ipynb](#)

### 3. Creating a series from Lists:

In order to create a series from list, we have to first create a list after that we can create a series from list.



The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with various icons: file, new, cut, copy, paste, up, down, run (which is highlighted with a red box), cell, code, and others. Below the toolbar, a code cell is shown with the following Python code:

```
In [4]: import pandas as pd

# a simple list
list = ['E', 'd', 't', 'e', 'c', 'H']

# create series form a List
ser = pd.Series(list)
print(ser)
```

The output of the code is displayed below the cell:

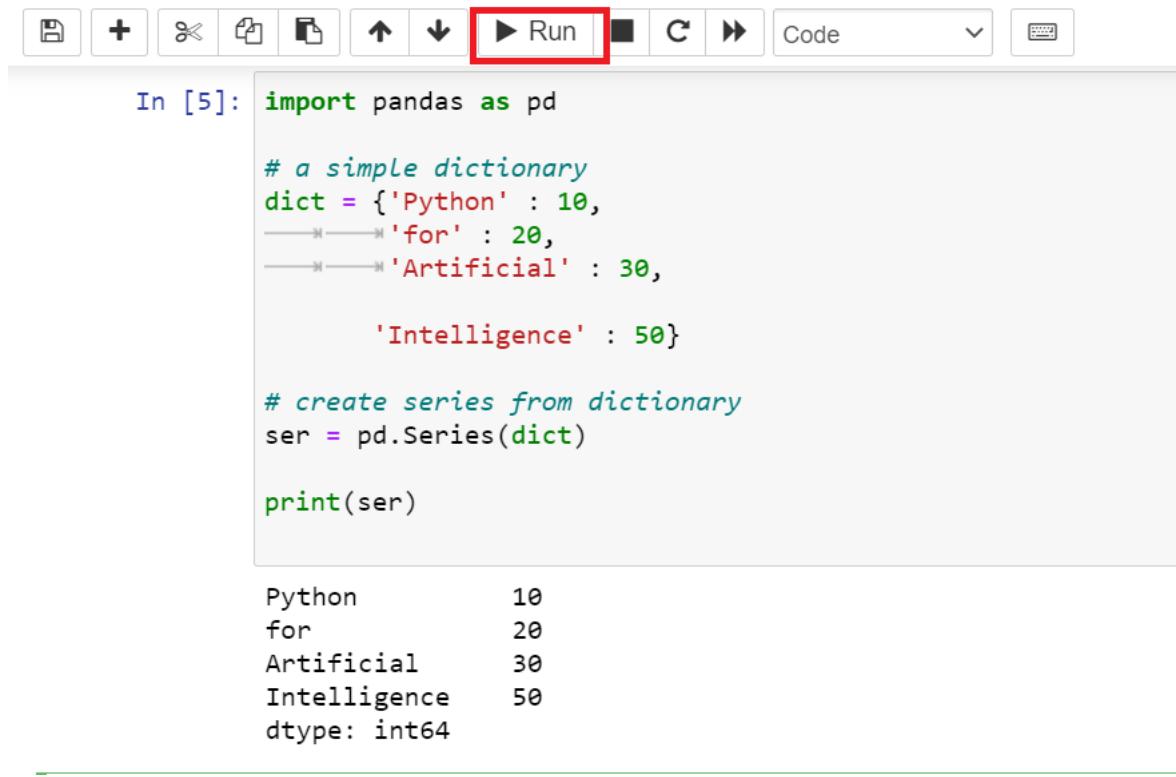
```
0    E
1    d
2    t
3    e
4    c
5    H
dtype: object
```

At the bottom, there's another empty code cell labeled "In [ ]:".

[Code: Creating Series from simple datatypes.ipynb](#)

### 4. Creating a series from Dictionary:

In order to create a series from dictionary, we have to first create a dictionary after that we can make a series using dictionary. Dictionary key are used to construct a index.



The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with various icons: file, new, cell, cut, copy, paste, up, down, run (which is highlighted with a red box), cell, code, and a keyboard icon. Below the toolbar, the code cell In [5] contains Python code to create a pandas Series from a dictionary. The output cell shows the resulting Series:

```
In [5]: import pandas as pd

# a simple dictionary
dict = {'Python' : 10,
         'for' : 20,
         'Artificial' : 30,
         'Intelligence' : 50}

# create series from dictionary
ser = pd.Series(dict)

print(ser)
```

```
Python    10
for      20
Artificial    30
Intelligence    50
dtype: int64
```

Code: Creating Series from simple datatypes.ipynb

## 3.24 Data Storage Formats in Pandas

The different data storage formats available to be manipulated by Pandas library are text, binary and SQL. Below is a table containing available ‘readers’ and ‘writers’ functions of the pandas I/O API set with data format and description. [6.5]

Format Type	Data Description	Reader	Writer
text	<a href="#">CSV</a>	<code>read_csv</code>	<code>to_csv</code>
text	Fixed-Width Text File	<code>read_fwf</code>	
text	<a href="#">JSON</a>	<code>read_json</code>	<code>to_json</code>
text	<a href="#">HTML</a>	<code>read_html</code>	<code>to_html</code>
text	Local clipboard	<code>read_clipboard</code>	<code>to_clipboard</code>
binary	<a href="#">MS Excel</a>	<code>read_excel</code>	<code>to_excel</code>
binary	<a href="#">OpenDocument</a>	<code>read_excel</code>	
binary	<a href="#">HDF5 Format</a>	<code>read_hdf</code>	<code>to_hdf</code>
binary	<a href="#">Feather Format</a>	<code>read_feather</code>	<code>to_feather</code>
binary	<a href="#">Parquet Format</a>	<code>read_parquet</code>	<code>to_parquet</code>
binary	<a href="#">ORC Format</a>	<code>read_orc</code>	

Format Type	Data Description	Reader	Writer
binary	<a href="#">Msgpack</a>	<a href="#">read_msgpack</a>	<a href="#">to_msgpack</a>
binary	<a href="#">Stata</a>	<a href="#">read_stata</a>	<a href="#">to_stata</a>
binary	<a href="#">SAS</a>	<a href="#">read_sas</a>	
binary	<a href="#">SPSS</a>	<a href="#">read_spss</a>	
binary	<a href="#">Python Pickle Format</a>	<a href="#">read_pickle</a>	<a href="#">to_pickle</a>
SQL	<a href="#">SQL</a>	<a href="#">read_sql</a>	<a href="#">to_sql</a>
SQL	<a href="#">Google BigQuery</a>	<a href="#">read_gbq</a>	<a href="#">to_gbq</a>

Figure: “readers” and “writers” functions in pandas

Reference: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/io.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html)

## 3.25 CSV file and JSON file

### What is CSV file?

A CSV is a comma-separated values file, which allows data to be saved in a tabular format. CSV files can be used with many spreadsheet program, such as Microsoft Excel or Google Spreadsheets. They differ from other spreadsheet file types because you can only have a single sheet in a file, they cannot save cell, column, or row. Also, you cannot save formulas in this format.

### Why are .CSV files used?

These files serve a number of different business purposes. Take, for instance, they help companies export a high volume of data to a more concentrated database.

They also serve two other primary business functions:

- CSV files are plain-text files, making them easier for the website developer to create
- Since they're plain text, they're easier to import into a spreadsheet or another storage database, regardless of the specific software you're using.
- To better organize large amounts of data.

### How do I save CSV files?

Saving CSV files is relatively easy, you just need to know where to change the file type. Under the "File name" section in the "Save As" tab, you can select "Save as type" and change it to "CSV (Comma delimited) (\*.csv)". Once that option is selected, you are on your way to quicker and easier data organization. This should be the same for both Apple and Microsoft operating systems.

## What is a JSON file?

A JSON file is a file that stores simple data structures and objects in JavaScript Object Notation (JSON) format, which is a standard data interchange format. It is primarily used for transmitting data between a web application and a server. JSON files are lightweight, text-based, human-readable, and can be edited using a text editor.

## How do I open a JSON file?

Because JSON files are plain text files, you can open them in any text editor, including:

- Microsoft Notepad (Windows)
- AppleTextEdit (Mac)
- Vim (Linux)
- GitHub Atom (cross-platform)

You can also open a JSON file in the Google Chrome and Mozilla Firefox web browsers by dragging and dropping the file into your browser window.

## Structures of JSON

JSON supports two widely used (amongst programming languages) data structures.

- **A collection of name/value pairs.** Different programming languages support this data structure in different names. Like object, record, struct, dictionary, hash table, keyed list, or associative array.
- **An ordered list of values.** In various programming languages, it is called as array, vector, list, or sequence.

Since data structure supported by JSON is also supported by most of the modern programming languages, it makes JSON a very useful data-interchange format.

Syntax:

{ string : value, .....}

### *Explanation of Syntax*

An object starts and ends with '{' and '}'. Between them, a number of string value pairs can reside. String and value is separated by a ':' and if there are more than one string value pairs, they are separated by ','.

### Example

```
{  
  "firstName": "John",  
  "lastName": "Maxwell",  
  "age": 40,  
  "email": "john@example.com"  
}
```

In JSON, objects can nest arrays (starts and ends with '[' and ']') within it. The following example shows that.

```
{  
  "Students": [  
    { "Name": "Amit Goenka" ,  
      "Major": "Physics" },  
    { "Name": "Smita Pallod" ,  
      "Major": "Chemistry" },  
    { "Name": "Rajeev Sen" ,  
      "Major": "Mathematics" }  
  ]  
}
```

### Array:

Syntax:

```
[ value, .....]
```

### *Explanation of Syntax:*

An Array starts and ends with '[' and ']'. Between them, a number of values can reside. If there are more than one values, they are separated by ','.

Example

```
[100, 200, 300, 400]
```

If the JSON data describes an array, and each element of that array is an object.

```
[  
  {  
    "name": "John Maxwell",  
    "email": "john@example.com"  
  },  
  {  
    "name": "Dale Carnegie",  
    "email": "dale@example.com"  
  }  
]
```

Remember that even arrays can also be nested within an object. The following shows that.

```
{  
  "firstName": "John",  
  "lastName": "Maxwell",  
  "age": 40,  
  "address":  
    {  
      "streetAddress": "144 J B Queens Road",  
      "city": "Dallas",  
      "state": "Washington",  
      "postalCode": "75001"  
    },  
  "phoneNumber":  
    [  
      {  
        "type": "personal",  
        "number": "(214)5096995"  
      },  
      {  
        "type": "fax",  
        "number": "13235551234"  
      }  
    ]  
}
```

### Value

Syntax:

String || Number || Object || Array || TRUE || FALSE || NULL

A value can be a string, a number, an object, an Array, a Boolean value (i.e. true or false) or Null. This structure can be nested.

### String

A string is a sequence of zero or more Unicode characters, enclosed by double quotes, using backslash escapes. A character is represented as a single character string, similar to a C or Java string.

The following table shows supported string types.

String Types	Description
"	A double quotation mark.
\	Reverse Solidus
/	Solidus

b	Backspace
f	form feed
n	newline
r	Carriage return
t	Horizontal tab
u	Four hexadecimal digits

### ***Number***

The following table shows supported number types.

Number Types	Description
Integer	Positive or negative Digits.1-9 and 0.
Fraction	Fractions like .8.
Exponent	e, e+, e-, E, E+, E-

### ***Whitespace***

Whitespace can be placed between any pair of supported data-types.

## **3.26 Reading data from files**

### **Load CSV files to Python Pandas**

The basic process of loading data from a CSV file into a Pandas DataFrame is achieved using the “read\_csv” function in Pandas:

```
# Load the Pandas libraries with alias 'pd'  
import pandas as pd
```

```
# Read data from file 'filename.csv'  
# (in the same directory that your python process is based)  
# Control delimiters, rows, column names with read_csv (see later)  
data = pd.read_csv("filename.csv")
```

```
# Preview the first 5 lines of the loaded data  
data.head();
```

While this code seems simple, an understanding of three fundamental concepts is required to fully grasp and debug the operation of the data loading procedure if you run into issues:

#### ***File Extensions and File Types***

The first step to working with comma-separated-value (CSV) files is understanding the concept of file types and file extensions.

1. Data is stored on your computer in individual “files”, or containers, each with a different name.
2. Each file contains data of different types – the internals of a Word document is quite different from the internals of an image.
3. Computers determine how to read files using the “file extension”, that is the code that follows the dot (“.”) in the filename.
4. So, a filename is typically in the form “<random name>.<file extension>”.  
Examples:
  - project1.DOCX – a Microsoft Word file called project1.
  - shanes\_file.TXT – a simple text file called shanes\_file
  - IMG\_5673.JPG – An image file called IMG\_5673.
  - Other well known file types and extensions include: XLSX: Excel, PDF: Portable Document Format, PNG – images, ZIP – compressed file format, GIF – animation, MPEG – video, MP3 – music etc. See a complete list of extensions [here](#).
5. A CSV file is a file with a “.csv” file extension, e.g. “data.csv”, “super\_information.csv”. The “CSV” in this case lets the computer know that the data contained in the file is in “comma separated value” format.

### ***Data Representation in CSV files***

A “CSV” file, that is, a file with a “csv” filetype, is a basic text file. Any text editor such as NotePad on windows or TextEdit on Mac, can open a CSV file and show the contents. Sublime Text is a wonderful and multi-functional text editor option for any platform.

CSV is a standard form for storing tabular data in text format, where commas are used to separate the different columns, and newlines (carriage return / press enter) are used to separate rows. Typically, the first row in a CSV file contains the names of the columns for the data.

An example of a table data set and the corresponding CSV-format data is shown in the diagram below.

Data opened in Microsoft Excel Spreadsheet

	A	B	C	D
1	Name	Age	Country	Goals
2	James Rodriguez		26 Colombia	6
3	Thomas Muller		28 Germany	5
4	Lionel Messi		31 Argentina	4
5	Neymar		26 Brazil	4
6	Robin van Persie		34 Netherlands	4

Data in CSV format opened in Text Editor

```

1 Name,Age,Country,Goals
2 James Rodriguez,26,Colombia,6
3 Thomas Muller,28,Germany,5
4 Lionel Messi,31,Argentina,4
5 Neymar,26,Brazil,4
6 Robin van Persie,34,Netherlands,4
7
  
```

Column names are first row in text file

Commas are used to separate columns

Figure: Comma-separated value files, or CSV files, are simple text files where commas and newlines are used to define tabular data in a structured way.

Reference: <https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files>

Note that almost any tabular data can be stored in CSV format – the format is popular because of its simplicity and flexibility. You can create a text file in a text editor, save it with a .csv extension, and open that file in Excel or Google Sheets to see the table form.

### Other Delimiters / Separators – TSV files

The comma separation scheme is by far the most popular method of storing tabular data in text files.

The choice of the ‘,’ comma character to delimit columns, however, is arbitrary, and can be substituted where needed. Popular alternatives include tab (“\t”) and semi-colon (“;”). Tab-separate files are known as TSV (Tab-Separated Value) files. When loading data with Pandas, the `read_csv` function is used for reading any delimited text file, and by changing the delimiter using the `sep` parameter.

### Delimiters in Text Fields – Quotefchar

One complication in creating CSV files is if you have commas, semicolons, or tabs actually in one of the text fields that you want to store. In this case, it's important to use a “quote character” in the CSV file to create these fields.

The quote character can be specified in `Pandas.read_csv` using the `quotefchar` argument. By default (as with many systems), it's set as the standard quotation marks (“”). Any commas (or other delimiters as demonstrated below) that occur between two quote characters will be ignored as column separators.

In the example shown, a semicolon-delimited file, with quotation marks as a quotechar is loaded into Pandas, and shown in Excel. The use of the quotechar allows the “NickName” column to contain semicolons without being split into more columns.

**Semi-colon separated data in text file**

```
CustomerId; CustomerName; Address; Age; NickNames
1;Shane Lynn;Dublin, Ireland; 30;"Shaneo;Lynno;Slynn"
2;Johnny Ives;London, United Kingdom;40;"Johnson;Big John;Ivy"
3;Simon Smith;Rue de Rue, Paris, France;50;"Frenchy;Smitho;Hammer"
4;Ronald Mc Donald;The big Farm, McDonalds Farm; 60;"Ronnie;Maccie;Donnie"
5;Jonathan Swift;Celbridge Abbey, Celbridge, Ireland;70;"Jonno;Speedy;Swifter"
```

Semicolons (;) are used here to separate columns

**Semi-colon separated data loaded into Excel**

A	B	C	D	E	
1	<b>CustomerId</b>	<b>CustomerName</b>	<b>Address</b>	<b>Age</b>	<b>NickNames</b>
2	1	Shane Lynn	Dublin, Ireland	30	Shaneo;Lynno;Slynn
3	2	Johnny Ives	London, United Kingdom	40	Johnson;Big John;Ivy
4	3	Simon Smith	Rue de Rue, Paris, France	50	Frenchy;Smitho;Hammer
5	4	Ronald Mc Donald	The big Farm, McDonalds Farm	60	Ronnie;Maccie;Donnie
6	5	Jonathan Swift	Celbridge Abbey, Celbridge, Ireland	70	Jonno;Speedy;Swifter

The data in the last column contains semicolons, so the quotation character is used to ‘quote’ the values

**Semi-colon separated data loaded to Pandas**

```
pd.read_csv('test_delimited.csv', sep=';', quotechar='"', encoding='utf8')
```

The 'sep' argument tells Pandas how to break up data into columns

CustomerID	CustomerName	Address	Age	NickNames	
0	1	Shane Lynn	Dublin, Ireland	30	Shaneo;Lynno;Slynn
1	2	Johnny Ives	London, United Kingdom	40	Johnson;Big John;Ivy
2	3	Simon Smith	Rue de Rue, Paris, France	50	Frenchy;Smitho;Hammer
3	4	Ronald Mc Donald	The big Farm, McDonalds Farm	60	Ronnie;Maccie;Donnie
4	5	Jonathan Swift	Celbridge Abbey, Celbridge, Ireland	70	Jonno;Speedy;Swifter

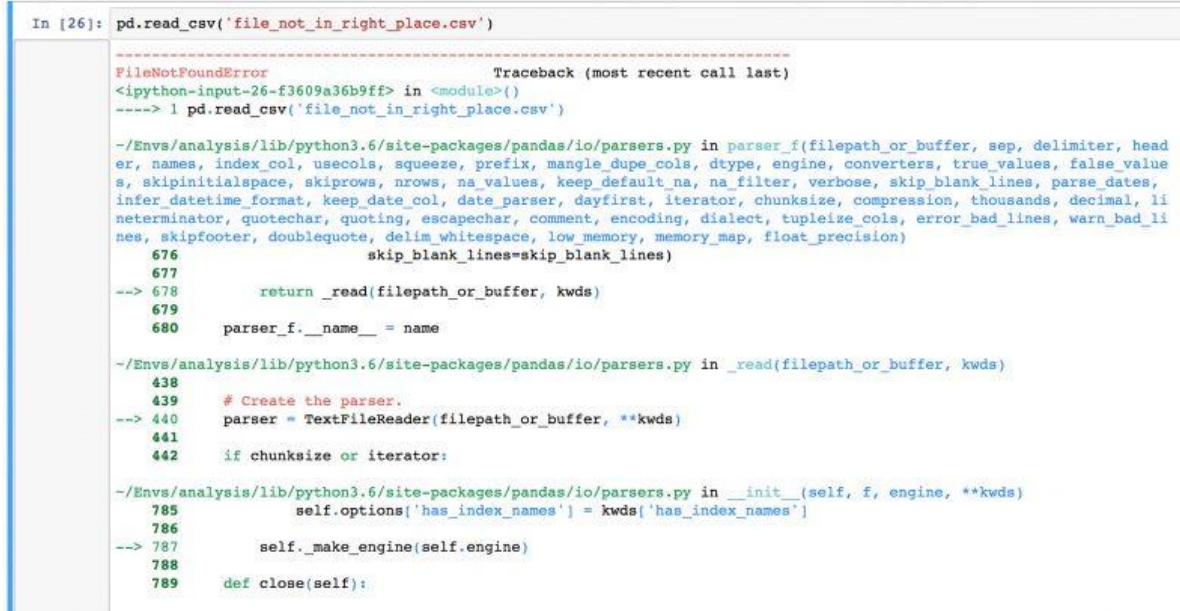
Specify the quotechar if necessary - the default is “”.

Figure: Other than commas in CSV files, Tab-separated and Semicolon-separated data is popular also. Quote characters are used if the data in a column may contain the separating character. In this case, the ‘NickName’ column contains semicolon characters, and so this column is “quoted”. Specify the separator and quote character in pandas.read\_csv

Reference: <https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files>

## Python – Paths, Folders, Files

When you specify a filename to Pandas.read\_csv, Python will look in your “current working directory”. Your working directory is typically the directory that you started from your Python process or Jupyter notebook.



In [26]: pd.read\_csv('file\_not\_in\_right\_place.csv')

```
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-26-f3609a36b9ff> in <module>()
      1 pd.read_csv('file_not_in_right_place.csv')

-/Envs/analysis/lib/python3.6/site-packages/pandas/io/parsers.py in parser_f(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirst, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, escapechar, comment, encoding, dialect, tupleize_cols, error_bad_lines, warn_bad_lines, skipfooter, doublequote, delim_whitespace, low_memory, memory_map, float_precision)
    676                     skip_blank_lines=skip_blank_lines)
    677
--> 678     return _read(filepath_or_buffer, kwds)
    679
    680     parser_f.__name__ = name

-/Envs/analysis/lib/python3.6/site-packages/pandas/io/parsers.py in _read(filepath_or_buffer, kwds)
    438
    439     # Create the parser.
--> 440     parser = TextFileReader(filepath_or_buffer, **kwds)
    441
    442     if chunksize or iterator:

-/Envs/analysis/lib/python3.6/site-packages/pandas/io/parsers.py in __init__(self, f, engine, **kwds)
    785         self.options['has_index_names'] = kwds['has_index_names']
    786
--> 787         self._make_engine(self.engine)
    788
    789     def close(self):
```

Figure: Pandas searches your ‘current working directory’ for the filename that you specify when opening or loading files. The FileNotFoundError can be due to a misspelled filename, or an incorrect working directory.

Reference: <https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files/>

### Finding your Python Path

Your Python path can be displayed using the built-in “**os**” module. The OS module is for operating system dependent functionality into Python programs and scripts.

To find your current working directory, the function required is **os.getcwd()**. The **os.listdir()** function can be used to display all files in a directory, which is a good check to see if the CSV file you are loading is in the directory as expected.

# Find out your current working directory

```
import os
print(os.getcwd())
# Out: /Users/shane/Documents/blog
# Display all of the files found in your current working directory
print(os.listdir(os.getcwd()))
# Out: ['test_delimited.ssv', 'CSV Blog.ipynb', 'test_data.csv']
In the example above, my current working directory is in the '/Users/Shane/Document/blog' directory. Any files that are places in this directory will be immediately available to the Python file_open() function or the Pandas read_csv() function.
```

Instead of moving the required data files to your working directory, you can also change your current working directory to the directory where the files reside using **os.chdir()**.

### File Loading: Absolute and Relative Paths

When specifying file names to the read\_csv function, you can supply both absolute or relative file paths.

- A relative path is the path to the file if you start from your current working directory. In relative paths, typically the file will be in a subdirectory of the working directory and the path will not start with a drive specifier, e.g. (data/test\_file.csv). The characters ‘..’ are used to move to a parent directory in a relative path.
- An absolute path is the complete path from the base of your file system to the file that you want to load, e.g. c:/Documents/Shane/data/test\_file.csv. Absolute paths will start with a drive specifier (c:/ or d:/ in Windows, or ‘/’ in Mac or Linux)

It's recommended and preferred to use relative paths where possible in applications, because absolute paths are unlikely to work on different computers due to different directory structures.

```
In [32]: # Show Current working directory
import pandas as pd
import os
os.getcwd()

Out[32]: '/Users/shane/Documents/Blog/read_csv'

In [33]: os.listdir(os.getcwd())
Out[33]: ['.ipynb_checkpoints', 'test_delimited.ssv', 'CSV Blog.ipynb', 'test_data.csv']

In [34]: # load file using relative path
pd.read_csv('test_data.csv')

Out[34]:
   id  first_name  second_name
0   1      Shane       Lynn
1   2     Jimmy       Jacobs
2   3      Mark    Christos
3   4    Seamus     O'Higgins
4   5      Juän      Encoding

In [35]: # load file using absolute path (same result)
pd.read_csv('/Users/shane/Documents/Blog/read_csv/test_data.csv')

Out[35]:
   id  first_name  second_name
0   1      Shane       Lynn
1   2     Jimmy       Jacobs
2   3      Mark    Christos
3   4    Seamus     O'Higgins
4   5      Juän      Encoding
```

Figure: Loading the same file with Pandas read\_csv using relative and absolute paths. Relative paths are directions to the file starting at your current working directory, where absolute paths always start at the base of your file system.

Reference:<https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files>

## Pandas CSV File Loading Errors

The most common error's you'll get while loading data from CSV files into Pandas will be:

1. **FileNotFoundException:** File b'filename.csv' does not exist

A File Not Found error is typically an issue with path setup, current directory, or file name confusion (file extension can play a part here!)

2. **UnicodeDecodeError:** 'utf-8' codec can't decode byte in position : invalid continuation byte

A Unicode Decode Error is typically caused by not specifying the encoding of the file, and happens when you have a file with non-standard characters. For a quick fix, try opening the file in Sublime Text, and re-saving with encoding 'UTF-8'.

3. **pandas.parser.CParserError: Error tokenizing data.**

Parse Errors can be caused in unusual circumstances to do with your data format – try to add the parameter “engine='python” to the read\_csv function call; this changes the data reading function internally to a slower but more stable method.

## Advanced Read CSV Files

There are some additional flexible parameters in the Pandas read\_csv() function that are useful to have in your arsenal of data science techniques:

- Specifying Data Types

As mentioned before, CSV files do not contain any type information for data. Data types are inferred through examination of the top rows of the file, which can lead to errors. To manually specify the data types for different columns, the dtype parameter can be used with a dictionary of column names and data types to be applied, for example: dtype={"name": str, "age": np.int32}

Note that for dates and date times, the format, columns, and other behaviour can be adjusted using parse\_dates, date\_parser, dayfirst, keep\_date parameters.

- Skipping and Picking Rows and Columns From File

The nrows parameter specifies how many rows from the top of CSV file to read, which is useful to take a sample of a large file without loading completely. Similarly the skiprows parameter allows you to specify rows to leave out, either at the start of the file (provide an int), or throughout the file (provide a list of row indices). Similarly, the usecols parameter can be used to specify which columns in the data to load.

- Custom Missing Value Symbols

When data is exported to CSV from different systems, missing values can be specified with different tokens. The na\_values parameter allows you to customise the characters that are recognised as missing values. The default values interpreted as NA/NaN are: “, '#N/A', '#N/A N/A', '#NA', '-1.#IND', '-1.#QNAN', '-NaN', '-nan', '1.#IND', '1.#QNAN', 'N/A', 'NA', 'NULL', 'NaN', 'n/a', 'nan', 'null'.

### # Advanced CSV loading example

```
data = pd.read_csv(  
    "data/files/complex_data_example.tsv",      # relative python path to subdirectory
```

```
sep='\t'      # Tab-separated value file.  
quotechar='"',    # single quote allowed as quote character  
dtype={"salary": int},    # Parse the salary column as an integer  
usecols=['name', 'birth_date', 'salary']. # Only load the three columns specified.  
parse_dates=['birth_date'],   # Interpret the birth_date column as a date  
skiprows=10,     # Skip the first 10 rows of the file  
na_values=['.', '??']    # Take any '.' or '??' values as NA  
) [6.9]
```

## Load JSON files to Python Pandas

Below are the steps to load JSON String into Pandas DataFrame

### **Step 1: Prepare the JSON String**

To start with a simple example, let's say that you have the following data about different products and their prices:

Product	Price
Desktop Computer	700
Tablet	250
iPhone	800
Laptop	1200

This data can be captured as a JSON string:

```
{"Product":{"0":"Desktop  
Computer","1":"Tablet","2":"iPhone","3":"Laptop"},"Price":{"0":700,"1":250,"2":800,"3":  
1200}}
```

### **Step 2: Create the JSON File**

Once you have your JSON string ready, save it within a JSON file.

Alternatively, you can copy the JSON string into Notepad, and then save that file with a *.json* file extension.

For example, open Notepad, and then copy the JSON string into it:

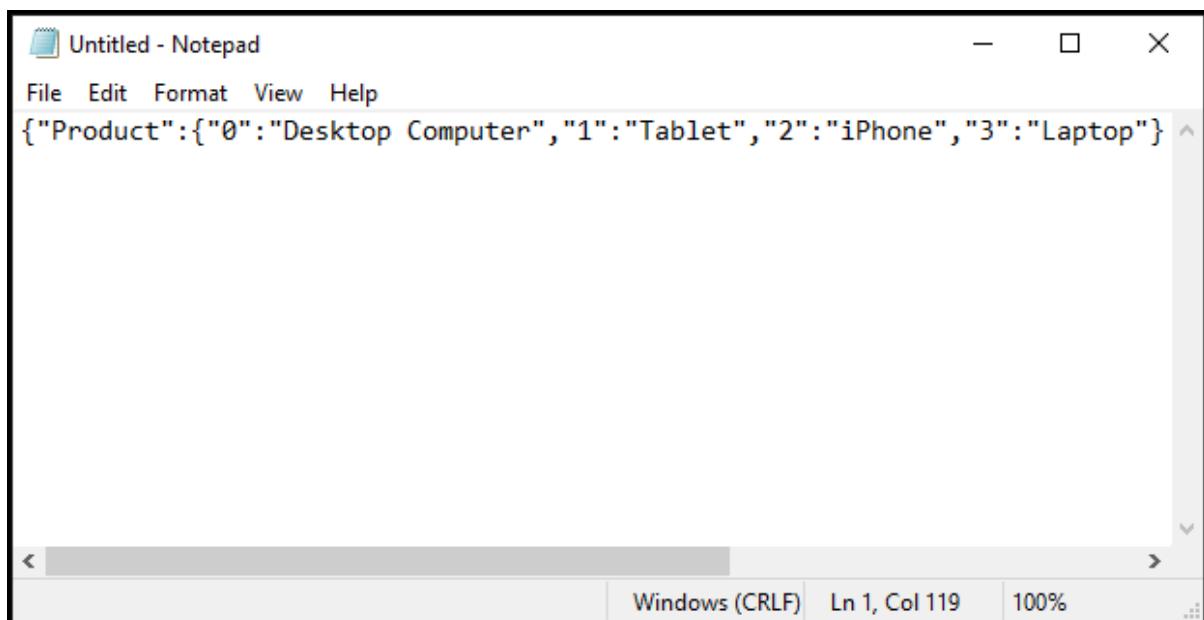


Figure: Copy the JSON string into the notepad.

Reference: <https://datatofish.com/load-json-pandas-dataframe>

Then, save the notepad with your desired file name and add the .json extension at the end of the file name. Here, I named the file as **data.json**:

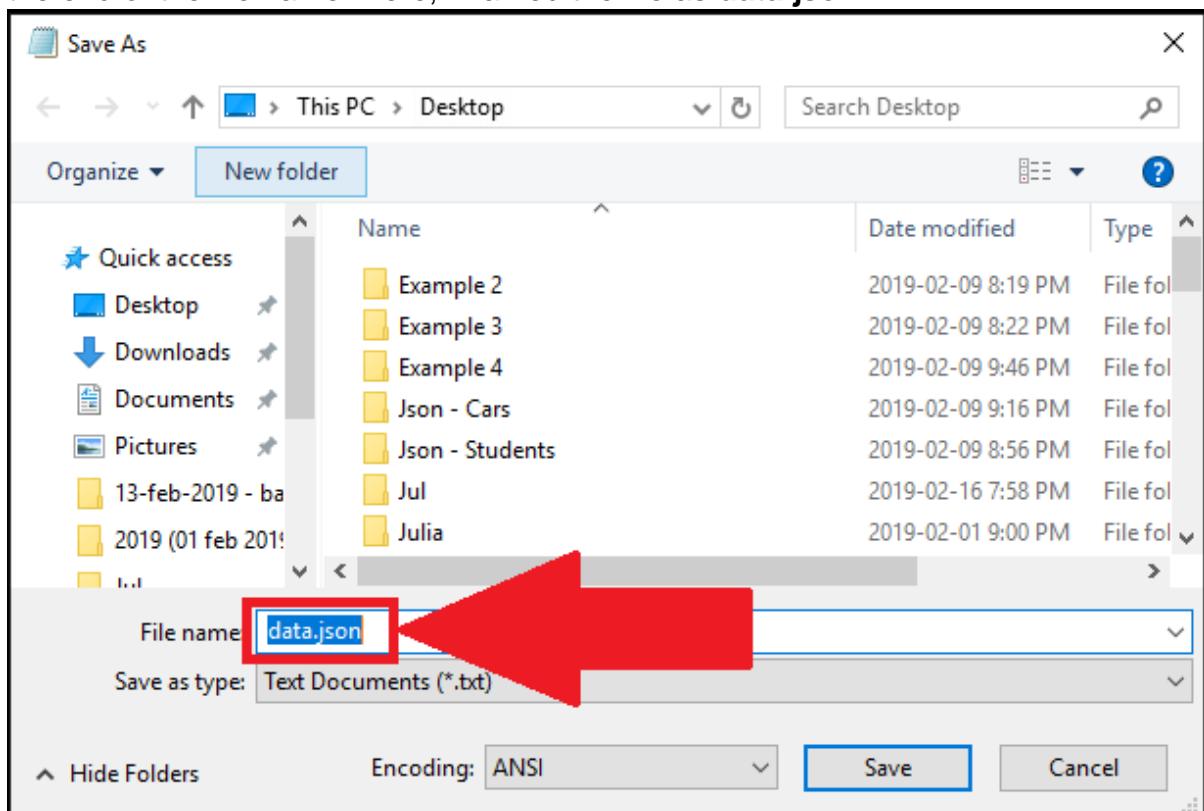


Figure: Saving the file with .json extension.

Reference: <https://datatofish.com/load-json-pandas-dataframe>

### Step 3: Load the JSON File into Pandas DataFrame

Finally, load your JSON file into Pandas DataFrame.

```
import pandas as pd
pd.read_json (r'Path where you saved the JSON file\File Name.json')
```

In this case, The JSON file is stored on the Desktop, under this path:

**C:\Users\Ron\Desktop\data.json**

So this is the code that is used to load the JSON file into the DataFrame:

```
import pandas as pd
df = pd.read_json (r'C:\Users\Ron\Desktop\data.json')
print (df)
```

Run the code in Python (adjusted to your path), and you'll get the following DataFrame:

	Product	Price
0	Desktop Computer	700
1	Tablet	250
2	iPhone	800
3	Laptop	1200

Figure: Output

### 3 different JSON strings

Below are 3 different ways that you could capture the data as JSON strings.

Each of those strings would generate a DataFrame with a different orientation when loading the files into Python.

#### 1. Index orientation

#### 2.

```
{"0":{"Product":"Desktop Computer","Price":700}, "1":{"Product":"Tablet","Price":250}, "2":{"Product":"iPhone","Price":800}, "3":{"Product":"Laptop","Price":1200}}
```

0	1	2	3
Price	700	250	800
Product	Desktop Computer	Tablet	iPhone
			Laptop

#### 3. Values orientation

```
[["Desktop Computer",700],["Tablet",250],["iPhone",800],["Laptop",1200]]
```

0	1
Desktop Computer	700
Tablet	250
iPhone	800
Laptop	1200

#### 4. Columns orientation

```
{"Product":{"0":"Desktop Computer","1":"Tablet","2":"iPhone","3":"Laptop"}, "Price":{"0":700,"1":250,"2":800,"3":1200}}
```

		Product	Price
0	Desktop Computer	700	
1	Tablet	250	
2	iPhone	800	
3	Laptop	1200	

## 3.27 Groupby Methods

Before applying groupby function to the dataset, let's go over a visual example. Assume we have two features. One is color which is a categorical feature and the other one is a numerical feature, values. We want to group values by color and calculate the mean (or any other aggregation) of values for different colors. Then finally sort the colors based on average values. The following figure shows the steps of this process.

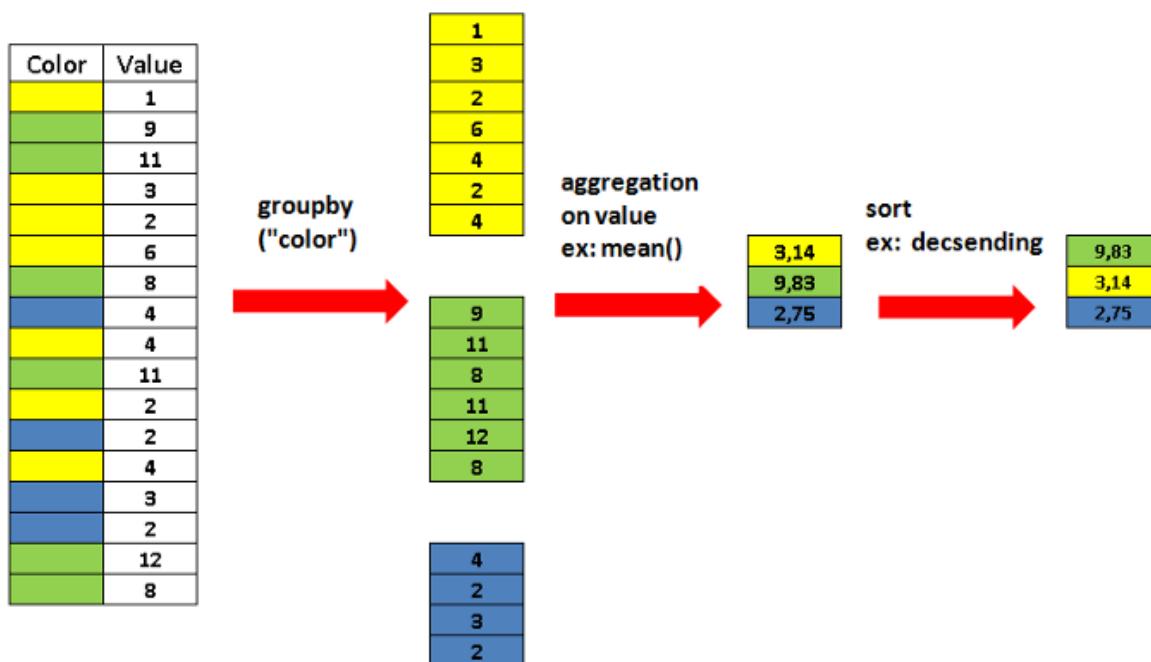


Figure: groupby in Pandas  
Reference:<https://towardsdatascience.com/pandas-groupby-explained-453692519d0>

## A Sample DataFrame

In order to demonstrate the effectiveness and simplicity of the grouping commands, we will need some data. The dataset contains 830 entries from my mobile phone log spanning a total time of 5 months. The CSV file can be loaded into a pandas DataFrame using the `pandas.DataFrame.from_csv()` function, and looks like this:

	date	duration	item	month	network	network_type
0	15/10/14 06:58	34.429	data	2014-11	data	data
1	15/10/14	13.000	call	2014-11	Vodafone	mobile

	06:58					
<b>2</b>	15/10/14 14:46	23.000	call	2014-11	Meteor	mobile
<b>3</b>	15/10/14 14:48	4.000	call	2014-11	Tesco	mobile
<b>4</b>	15/10/14 17:27	4.000	call	2014-11	Tesco	mobile
<b>5</b>	15/10/14 18:55	4.000	call	2014-11	Tesco	mobile
<b>6</b>	16/10/14 06:58	34.429	data	2014-11	data	data
<b>7</b>	16/10/14 15:01	602.000	call	2014-11	Three	mobile
<b>8</b>	16/10/14 15:12	1050.000	call	2014-11	Three	mobile
<b>9</b>	16/10/14 15:30	19.000	call	2014-11	voicemail	voicemail
<b>10</b>	16/10/14 16:21	1183.000	call	2014-11	Three	mobile
<b>11</b>	16/10/14 22:18	1.000	sms	2014-11	Meteor	mobile
...	...	...	...	...	...	...

The main columns in the file are:

1. date: The date and time of the entry
2. duration: The duration (in seconds) for each call, the amount of data (in MB) for each data entry, and the number of texts sent (usually 1) for each sms entry.
3. item: A description of the event occurring – can be one of call, sms, or data.
4. month: The billing month that each entry belongs to – of form 'YYYY-MM'.
5. network: The mobile network that was called/texted for each entry.
6. network\_type: Whether the number being called was a mobile, international ('world'), voicemail, landline, or other ('special') number.

The date column can be parsed using the **dateutil** library.

```
import pandas as pd
import dateutil

# Load data from csv file
data = pd.DataFrame.from_csv('phone_data.csv')
# Convert date from string to date times
data['date'] = data['date'].apply(dateutil.parser.parse, dayfirst=True)
```

## Summarizing the DataFrame

Once the data has been loaded into Python, Pandas makes the calculation of different statistics very simple. For example, mean, max, min, standard deviations and more for columns are easily calculable:

```
# How many rows the dataset
```

```
data['item'].count()
```

```
Out[38]: 830
```

```
# What was the longest phone call / data entry?
```

```
data['duration'].max()
```

```
Out[39]: 10528.0
```

```
# How many seconds of phone calls are recorded in total?
```

```
data['duration'][data['item'] == 'call'].sum()
```

```
Out[40]: 92321.0
```

```
# How many entries are there for each month?
```

```
data['month'].value_counts()
```

```
Out[41]:
```

```
2014-11    230
```

```
2015-01    205
```

```
2014-12    157
```

```
2015-02    137
```

```
2015-03    101
```

```
dtype: int64
```

```
# Number of non-null unique network entries
```

```
data['network'].nunique()
```

```
Out[42]: 9
```

The [.describe\(\)](#) function is a useful summarisation tool that will quickly display statistics for any variable or group it is applied to. The describe() output varies depending on whether you apply it to a numeric or character column.

## Groupby output format – Series or DataFrame?

The output from a groupby and aggregation operation varies between Pandas Series and Pandas Dataframes. As a rule of thumb, if you calculate more than one column of results, your result will be a DataFrame. For a single column of results, the agg function, by default, will produce a Series.

You can change this by selecting your operation column differently:

```
# produces Pandas Series
```

```
data.groupby('month')['duration'].sum()
```

```
# Produces Pandas DataFrame
```

```
data.groupby('month')[['duration']].sum()
```

The groupby output will have an index or [multi-index](#) on rows corresponding to your chosen grouping variables. To avoid setting this index, pass “as\_index=False” to the groupby operation.

```
data.groupby('month', as_index=False).agg({"duration": "sum"})
```

```
In [35]: data.groupby('month', as_index=False).agg({"duration": "sum"})
```

```
Out[35]:
```

	month	duration
0	2014-11	26639.441
1	2014-12	14641.870
2	2015-01	18223.299
3	2015-02	15522.299
4	2015-03	22750.441

Figure: Using the as\_index parameter while Grouping data in pandas prevents setting a row index on the result.

Reference: <https://www.shanelynn.ie/summarising-aggregation-and-grouping-data-in-python-pandas>

## Multiple Statistics per Group

The aggregation functionality provided by the **agg()** function allows multiple statistics to be calculated per group in one calculation.

### *Applying a single function to columns in groups*

Instructions for aggregation are provided in the form of a python dictionary or list. The dictionary keys are used to specify the columns upon which you'd like to perform operations, and the dictionary values to specify the function to run.

For example:

```
# Group the data frame by month and item and extract a number of stats from each group
data.groupby(
    ['month', 'item']
).agg(
{
    'duration':sum, # Sum duration per group
    'network_type': "count", # get the count of networks
    'date': 'first' # get the first date per group
})
```

## 3.28 Pivot Tables

You may be familiar with pivot tables in Excel to generate easy insights into your data. The function is quite similar to the group by function available in Pandas.

## How to Build a Pivot Table in Python

It's a table of statistics that helps summarize the data of a larger table by "pivoting" that data. In Pandas, we can construct a pivot table using the following syntax,  
`pandas.pivot_table(data, values=None, index=None, columns=None, aggfunc='mean', fill_value=None, margins=False, dropna=True, margins_name='All', observed=False)`

The levels in the pivot table will be stored in MultiIndex objects (hierarchical indexes) on the index and columns of the result DataFrame.

### Parameters

**Data: DataFrame**

**values: column to aggregate, optional**

**index: column, Grouper, array, or list of the previous**

If an array is passed, it must be the same length as the data. The list can contain any of the other types (except list). Keys to group by on the pivot table index. If an array is passed, it is being used as the same manner as column values.

**Columns: column, Grouper, array, or list of the previous**

If an array is passed, it must be the same length as the data. The list can contain any of the other types (except list). Keys to group by on the pivot table column. If an array is passed, it is being used as the same manner as column values.

**Aggfunc: function, list of functions, dict, default numpy.mean**

If list of functions passed, the resulting pivot table will have hierarchical columns whose top level are the function names (inferred from the function objects themselves) If dict is passed, the key is column to aggregate and value is function or list of functions.

**fill\_value: scalar, default None**

Value to replace missing values with (in the resulting pivot table, after aggregation).

**Margins: bool, default False**

Add all row / columns (e.g. for subtotal / grand totals).

**Dropna: bool, default True**

Do not include columns whose entries are all NaN.

**margins\_name: str, default 'All'**

Name of the row / column that will contain the totals when margins is True.

**Observed: bool, default False**

This only applies if any of the groupers are Categoricals. If True: only show observed values for categorical groupers. If False: show all values for categorical groupers.

**Returns: DataFrame**

An Excel style pivot table.

We'll use Pandas to import the data into a dataframe called df. We'll also print out the first five rows using the .head() function:

```
import pandas as pd
df = pd.read_excel('https://github.com/datagy/pivot_table_pandas/raw/master/sample_pivot.xlsx', parse_dates=['Date'])
print(df.head())
```

```
Out:
      Date    Region   Type  Units  Sales
0  2020-07-11    East Children's Clothing    18    306
1  2020-09-23   North Children's Clothing    14    448
2  2020-04-02   South Women's Clothing     17    425
3  2020-02-28    East Children's Clothing    26    832
4  2020-03-19   West  Women's Clothing      3     33
```

## Creating a Pivot Table in Pandas

We'll begin by aggregating the Sales values by the Region the sale took place in:

```
sales_by_region = pd.pivot_table(df, index = 'Region', values = 'Sales')
print(sales_by_region)
```

This returns the following output:

```
Sales    Region
East     408.182482
North    438.924051
South    432.956204
West     452.029412
```

This gave us a summary of the Sales field by Region. The default parameter for **aggfunc** is mean. Because of this, the Sales field in the resulting dataframe is the average of Sales per Region.

If we wanted to change the type of function used, we could use the **aggfunc** parameter. For example, if we wanted to return the sum of all Sales across a region, we could write:

```
total_by_region = pd.pivot_table(df, index = 'Region', values = 'Sales',
aggfunc='sum')
print(total_by_region)
```

This returns:

```
Sales    Region
East     167763
North    138700
South    59315
West     61476
```

## Filtering Python Pivot Tables

Let's create a dataframe that generates the **mean Sale** price by **Region**:

```
avg_region_price = pd.pivot_table(df, index = 'Region', values = 'Sales')
```

The values in this dataframe are:

Sales	Region
East	408.182482
North	438.924051
South	432.956204
West	452.029412

Now, say we wanted to filter the dataframe to only include Regions where the average sale price was over 450, we could write:

```
avg_region_price[avg_region_price['Sales'] > 450]
```

Sales	Region
West	452.029412

We can also apply multiple conditions, such as filtering to show only sales greater than 450 or less than 430.

```
avg_region_price[(avg_region_price['Sales'] > 450) | (avg_region_price['Sales'] < 430)]
```

We have wrapped each condition in brackets and separated the conditions by a **pipe** ( | ) symbol. This returns the following:

Sales	Region
East	408.182482
West	452.029412

## Adding Columns to a Pandas Pivot Table

Adding columns to a pivot table in Pandas can add another dimension to the tables. The **Columns** parameter allows us to add a key to aggregate by. For example, if we wanted to see the number of units sold by Type and by Region, we could write:

```
columns_example = pd.pivot_table(df, index = 'Type', columns = 'Region', values = 'Units', aggfunc = 'sum') print(columns_example)
```

Region	East	North	South	West
Type				
Children's Clothing	2318.0	1763.0	1017.0	789.0
Men's Clothing	2420.0	0.0	725.0	829.0
Women's Clothing	3372.0	2596.0	1056.0	1006.0

Columns are optional as we indicated above and provide the keys by which to separate the data. The pivot table aggregates the values in the values parameter.

## 3.29 Pandas Plotting

### Plot a Scatter Diagram using Pandas

Scatter plots are used to depict a relationship between two variables. Here are the steps to plot a scatter diagram using Pandas.

#### *Step 1: Prepare the data*

To start, prepare the data for your scatter diagram.

For example, the following data will be used to create the scatter diagram. This data captures the relationship between two variables related to an economy:

Unemployment_Rate	Stock_Index_Price
6.1	1500
5.8	1520
5.7	1525
5.7	1523
5.8	1515
5.6	1540
5.5	1545
5.3	1560
5.2	1555
5.2	1565

#### *Step 2: Create the DataFrame*

Once you have your data ready, you can proceed to create the DataFrame in Python. For our example, the DataFrame would look like this:

```
import pandas as pd
```

```
data = {'Unemployment_Rate': [6.1, 5.8, 5.7, 5.7, 5.8, 5.6, 5.5, 5.3, 5.2, 5.2],  
       'Stock_Index_Price': [1500, 1520, 1525, 1523, 1515, 1540, 1545, 1560, 1555, 1565]  
     }
```

```
df = pd.DataFrame(data, columns=['Unemployment_Rate', 'Stock_Index_Price'])  
print(df)
```

Run the code in Python, and you'll get the following DataFrame:

	Unemployment_Rate	Stock_Index_Price
0	6.1	1500
1	5.8	1520
2	5.7	1525
3	5.7	1523
4	5.8	1515
5	5.6	1540
6	5.5	1545
7	5.3	1560
8	5.2	1555
9	5.2	1565

### **Step 3: Plot the DataFrame using Pandas**

Finally, you can plot the DataFrame by adding the following syntax:

```
df.plot(x='Unemployment_Rate', y='Stock_Index_Price', kind = 'scatter')
```

Notice that you can specify the type of chart by setting **kind = 'scatter'**

You'll also need to add the Matplotlib syntax to show the plot (ensure that the Matplotlib package is install in Python):

- `import matplotlib.pyplot as plt`
- `plt.show()`

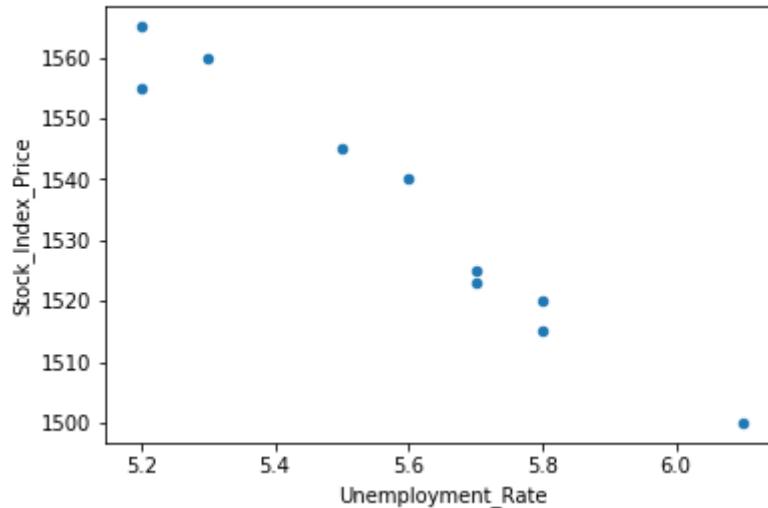
Putting everything together:

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
data = {'Unemployment_Rate': [6.1,5.8,5.7,5.7,5.8,5.6,5.5,5.3,5.2,5.2],
        'Stock_Index_Price': [1500,1520,1525,1523,1515,1540,1545,1560,1555,1565]
       }
```

```
df = pd.DataFrame(data,columns=['Unemployment_Rate','Stock_Index_Price'])
df.plot(x='Unemployment_Rate', y='Stock_Index_Price', kind = 'scatter')
plt.show();
```

Once you run the above code, you'll get the following scatter diagram:



## Plot a Line Chart using Pandas

Line charts are often used to display trends overtime. Let's now see the steps to plot a line chart using Pandas.

### Step 1: Prepare the data

To start, prepare your data for the line chart. Here is an example of a dataset that captures the unemployment rate over time:

Year	Unemployment_Rate
1920	9.8
1930	12
1940	8
1950	7.2
1960	6.9
1970	7
1980	6.5
1990	6.2
2000	5.5
2010	6.3

### Step 2: Create the DataFrame

Now create the DataFrame based on the above data:

```
import pandas as pd
data = {'Year': [1920, 1930, 1940, 1950, 1960, 1970, 1980, 1990, 2000, 2010],
        'Unemployment_Rate': [9.8, 12, 8, 7.2, 6.9, 7, 6.5, 6.2, 5.5, 6.3]}
df = pd.DataFrame(data, columns=['Year', 'Unemployment_Rate'])
print(df);
```

This is how the DataFrame would look like:

	Year	Unemployment_Rate
0	1920	9.8
1	1930	12.0
2	1940	8.0
3	1950	7.2
4	1960	6.9
5	1970	7.0
6	1980	6.5
7	1990	6.2
8	2000	5.5
9	2010	6.3

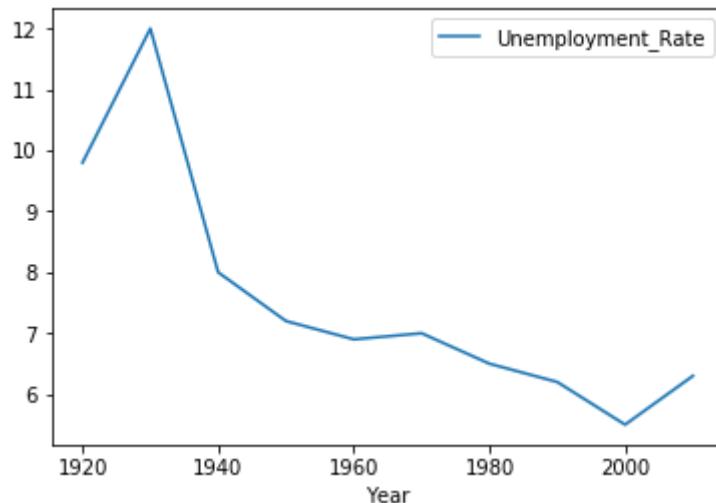
### Step 3: Plot the DataFrame using Pandas

Finally, plot the DataFrame by adding the following syntax:

```
df.plot(x ='Year', y='Unemployment_Rate', kind = 'line')
```

You'll notice that the `kind` is now set to '`line`' in order to plot the line chart.

```
import pandas as pd
import matplotlib.pyplot as plt
data = {'Year': [1920,1930,1940,1950,1960,1970,1980,1990,2000,2010],
        'Unemployment_Rate': [9.8,12,8,7.2,6.9,7,6.5,6.2,5.5,6.3]}
df = pd.DataFrame(data,columns=['Year','Unemployment_Rate'])
df.plot(x ='Year', y='Unemployment_Rate', kind = 'line')
plt.show();
```



### Plot a Bar Chart using Pandas

Bar charts are used to display categorical data. Let's now see how to plot a bar chart using Pandas.

### Step 1: Prepare your data

As before, you'll need to prepare your data. Here, the following dataset will be used to create the bar chart:

Country	GDP_Per_Capita
USA	45000
Canada	42000
Germany	52000
UK	49000
France	47000

### Step 2: Create the DataFrame

Create the DataFrame as follows:

```
import pandas as pd
data = {'Country': ['USA','Canada','Germany','UK','France'],
        'GDP_Per_Capita': [45000,42000,52000,49000,47000] }
df = pd.DataFrame(data,columns=['Country','GDP_Per_Capita'])
print (df);
```

You'll then get this DataFrame:

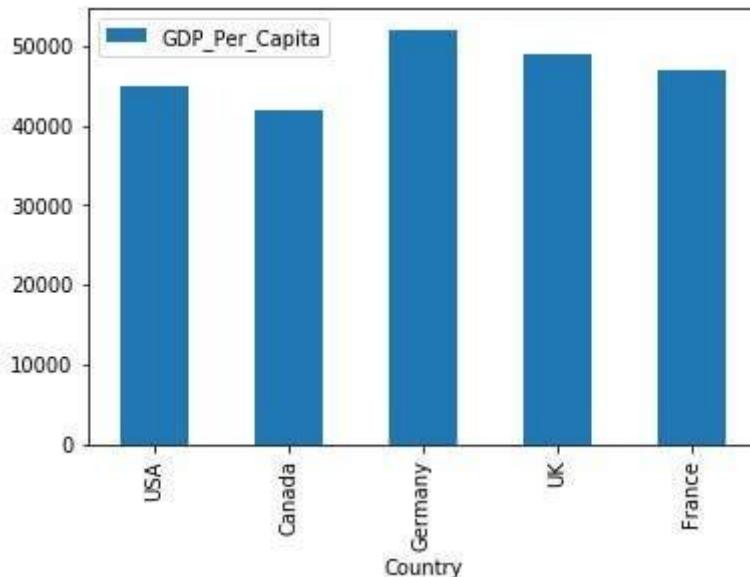
```
   Country  GDP_Per_Capita
0      USA            45000
1    Canada           42000
2  Germany           52000
3       UK            49000
4     France          47000
```

### Step 3: Plot the DataFrame using Pandas

Finally, add the following syntax to the Python code:

```
df.plot(x ='Country', y='GDP_Per_Capita', kind = 'bar')
In this case, set the kind = 'bar' to plot the bar chart.
import pandas as pd
import matplotlib.pyplot as plt
data = {'Country': ['USA','Canada','Germany','UK','France'],
        'GDP_Per_Capita': [45000,42000,52000,49000,47000]
       }
df = pd.DataFrame(data,columns=['Country','GDP_Per_Capita'])
df.plot(x ='Country', y='GDP_Per_Capita', kind = 'bar')
plt.show();
```

Run the code and you'll get this bar chart:



## Plot a Pie Chart using Pandas

### Step 1: Prepare your data

For demonstration purposes, the following data about the status of *tasks* was prepared:

<b>Tasks Pending</b>	300
<b>Tasks Ongoing</b>	500
<b>Tasks Completed</b>	700

The goal is to create a pie chart based on the above data.

### Step 2: Create the DataFrame

You can then create the DataFrame using this code:

```
import pandas as pd
data = {'Tasks': [300,500,700]}
df = pd.DataFrame(data,columns=['Tasks'],index = ['Tasks Pending','Tasks Ongoing','Tasks Completed'])
print (df);
```

You'll now see this DataFrame:

	Tasks
Tasks Pending	300
Tasks Ongoing	500
Tasks Completed	700

### Step 3: Plot the DataFrame using Pandas

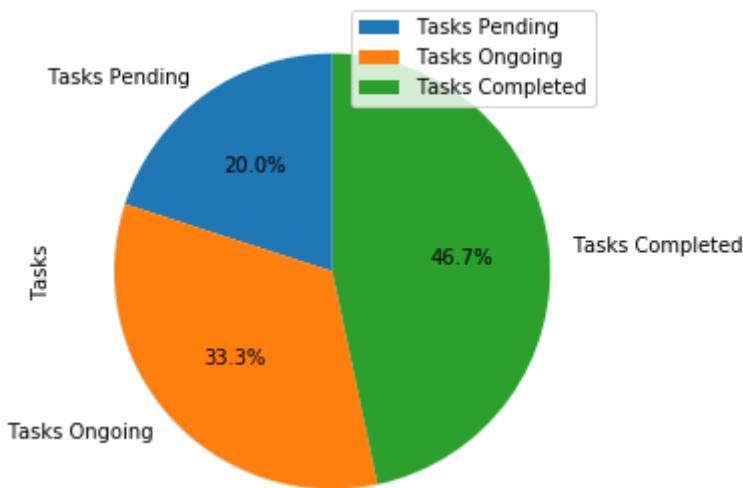
Finally, plot the DataFrame by adding the following syntax:

```
df.plot.pie(y='Tasks',figsize=(5, 5),autopct='%.1f%%', startangle=90)
```

The complete code is as follows:

```
import pandas as pd
import matplotlib.pyplot as plt
data = {'Tasks': [300,500,700]}
df = pd.DataFrame(data,columns=['Tasks'],index = ['Tasks Pending','Tasks Ongoing','Tasks Completed'])
df.plot.pie(y='Tasks',figsize=(5, 5), autopct='%.1f%%', startangle=90)
plt.show();
```

Once you run the code, you'll get this pie chart: **[6.15]**



So, in this chapter we have explored about various important concepts of Data analytics and the libraries which are used in data analysis like NumPy, Pandas and Matplotlib. Using these libraries, we can analyse our data and make sense out of data.

Please refer this [link \[Click Here\]](#) for all the codes/practical in this chapter.

# Chapter 4 : Building Machine Learning Models

## Learning Outcomes:

- Understand the basics of machine learning , its types and applications
- Create Machine Learning model using various Algorithms
- Demonstrate working of ML model without coding
- Able to differentiate between Supervised and Unsupervised Learning
- Able to identify and apply specific ML algorithm to solve real life problems

## 4.1 Machine Learning Basics

Machine Learning is undeniably one of the most influential and powerful technologies in today's world. Machine learning is a tool for turning information into knowledge. In the past 50 years, there has been an explosion of data. This mass of data is useless unless we analyze it and find the patterns hidden within. Machine learning techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover. The hidden patterns and knowledge about a problem can be used to predict future events and perform all kinds of complex decision making. 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.

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.

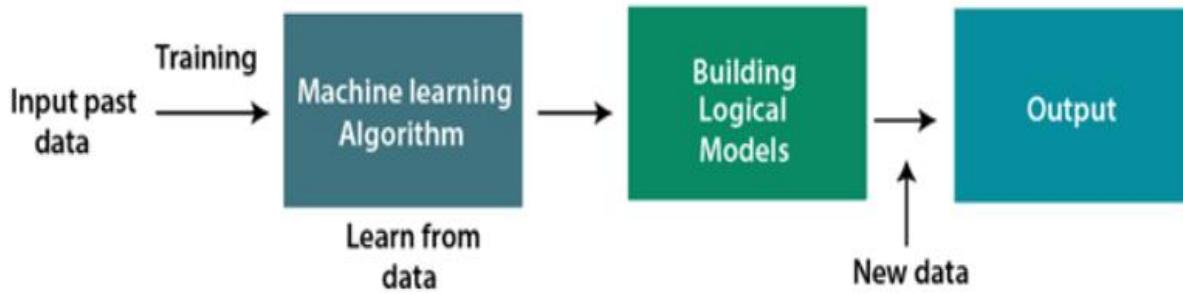


Image: ML working concept

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

## 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.

## 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 actually 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.

Let us see some real life applications of machine learning in our day-to-day life.

## Real time Application 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.

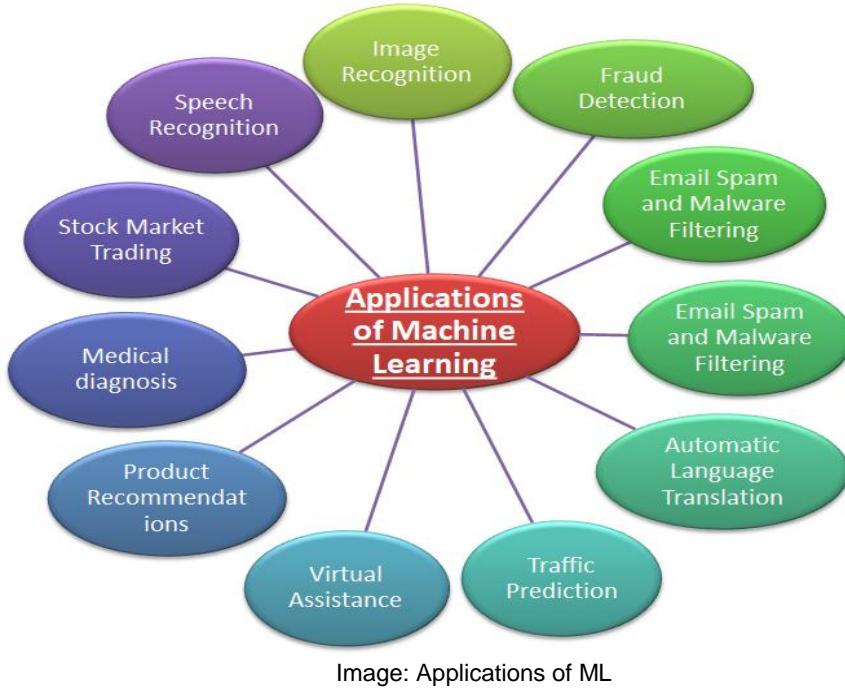


Image: Applications of ML

### 1. Image recognition

Image recognition is a well-known and widespread example of machine learning in the real world. It can identify an object as a digital image, based on the intensity of the pixels in black and white images or colour images.

- Real-world examples of image recognition:
- Label an x-ray as cancerous or not
- Assign a name to a photographed face (aka “tagging” on social media)
- Recognise handwriting by segmenting a single letter into smaller images
- Machine learning is also frequently used for facial recognition within an image. Using a database of people, the system can identify commonalities and match them to faces. This is often used in law enforcement.

### 2. Speech recognition

Machine learning can translate speech into text. Certain software applications can convert live voice and recorded speech into a text file. The speech can be segmented by intensities on time-frequency bands as well.

Real-world examples of speech recognition:

- Voice search
- Voice dialling

- Appliance control

Some of the most common uses of speech recognition software are devices like Google Home or Amazon Alexa.

### 3. Medical diagnosis

Machine learning can help with the diagnosis of diseases. Many physicians use chatbots with speech recognition capabilities to discern patterns in symptoms.

Real-world examples for medical diagnosis:

- Assisting in formulating a diagnosis or recommends a treatment option
- Oncology and pathology use machine learning to recognise cancerous tissue
- Analyse bodily fluids
- In the case of rare diseases, the joint use of facial recognition software and machine learning helps scan patient photos and identify phenotypes that correlate with rare genetic diseases.

### 4. Statistical arbitrage

Arbitrage is an automated trading strategy that's used in finance to manage a large volume of securities. The strategy uses a trading algorithm to analyze a set of securities using economic variables and correlations.

Real-world examples of statistical arbitrage:

- Algorithmic trading which analyses a market microstructure
- Analyse large data sets
- Identify real-time arbitrage opportunities

Machine learning optimizes the arbitrage strategy to enhance results.

### 5. Predictive analytics

Machine learning can classify available data into groups, which are then defined by rules set by analysts. When the classification is complete, the analysts can calculate the probability of a fault.

Real-world examples of predictive analytics:

- Predicting whether a transaction is fraudulent or legitimate
- Improve prediction systems to calculate the possibility of fault

Predictive analytics is one of the most promising examples of machine learning. It's applicable for everything; from product development to real estate pricing.

## 6. Extraction

Machine learning can extract structured information from unstructured data. Organizations amass huge volumes of data from customers. A machine learning algorithm automates the process of annotating datasets for predictive analytics tools.

Real-world examples of extraction:

- Generate a model to predict vocal cord disorders
- Develop methods to prevent, diagnose, and treat the disorders
- Help physicians diagnose and treat problems quickly

Typically, these processes are tedious. But machine learning can track and extract information to obtain billions of data samples.

Let us see the different techniques in machine learning.

## 4.2 Techniques of ML

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

### 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.

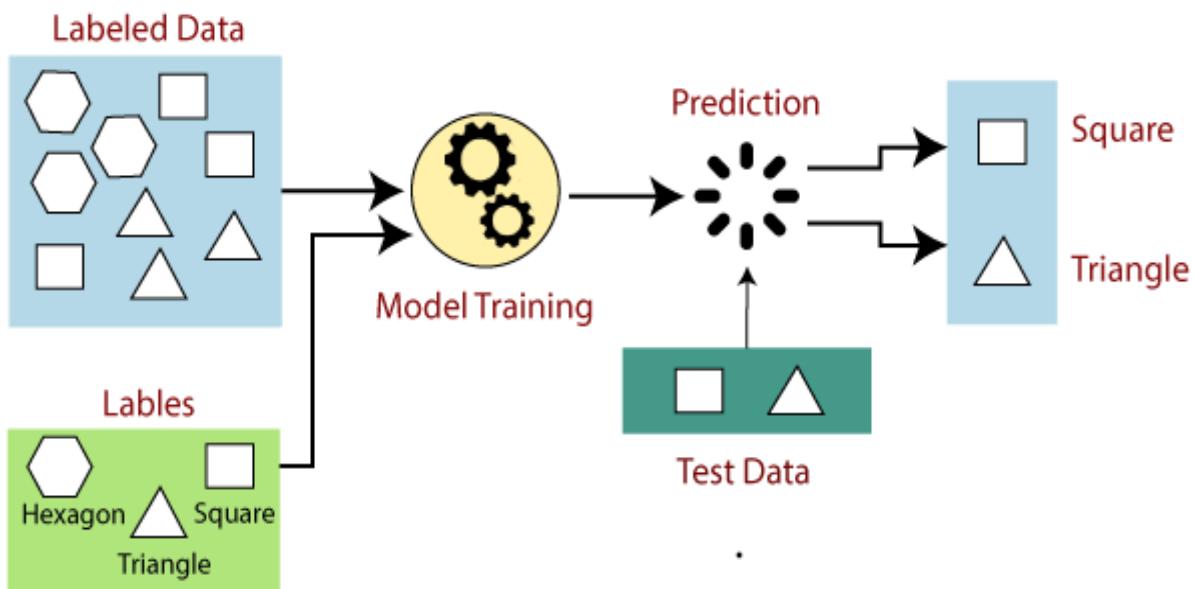


Image: Working of Supervised learning

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:

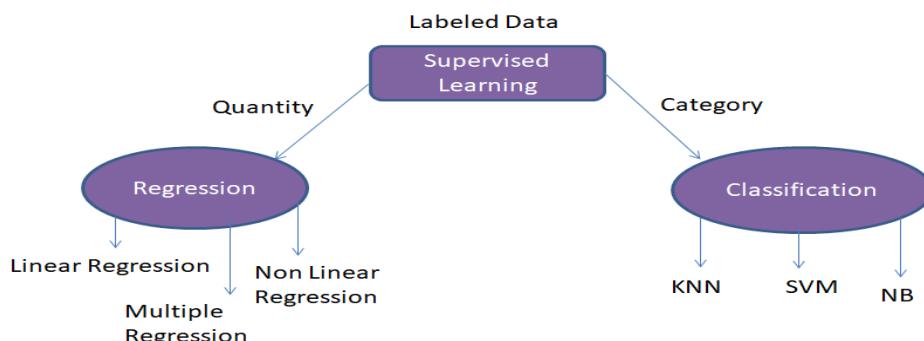


Image: Categories of supervised learning

## ***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.

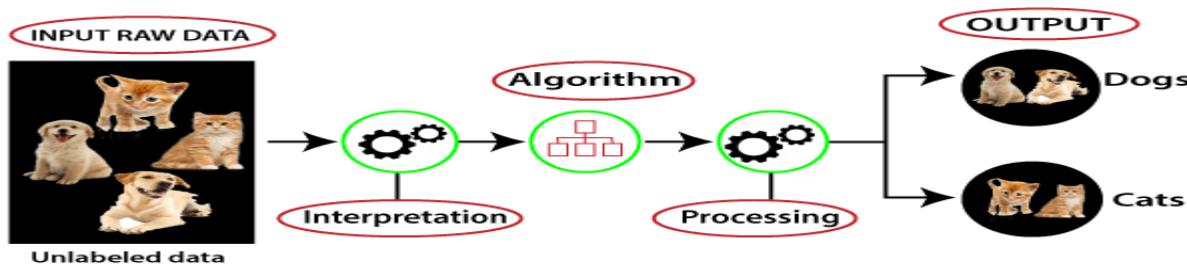


Image: Working of unsupervised learning  
 Reference: <https://www.javatpoint.com/unsupervised-machine-learning>

#### Types of Unsupervised Learning Algorithm:

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

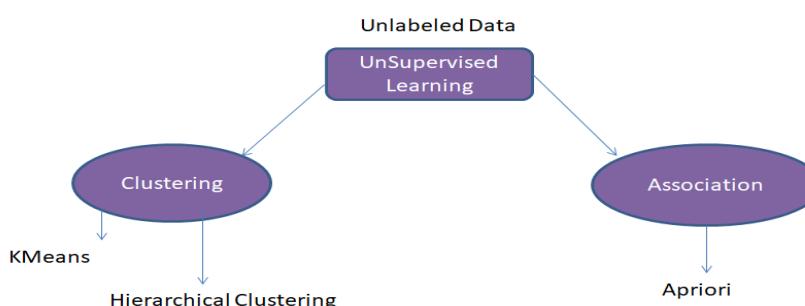


Image: Categories of unsupervised learning

#### Reinforcement Machine Learning

In this type of machine learning, during the training of algorithm it uses system of rewards and penalty. The learning system, called agent in this context, learns with an interactive environment. The agent selects and performs actions and receives rewards by performing correctly and penalties for performing incorrectly. In

reinforcement learning the agent learns by itself, without the intervention from a human. For example, Self-Driving Cars, robots etc.

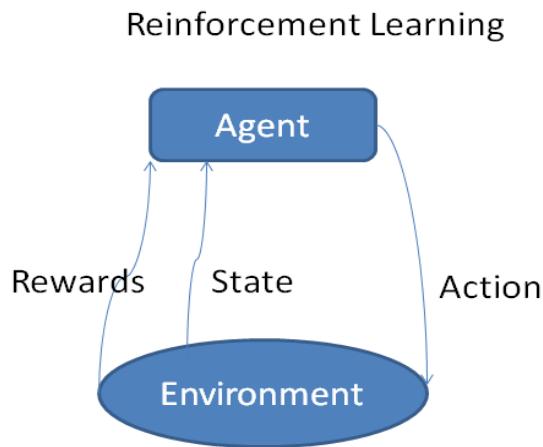


Image: Reinforcement Machine Learning

Now, after understanding the brief overview of machine learning and its types, let us explore a very popular library in machine learning which is used by many machine learning engineers and data scientists to perform various data science and AI projects, named as Scikit-learn. Let us get started with Scikit learn library.

## 4.3 Scikit Learn library overview

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

### ***Origin of Scikit-Learn***

It was originally called scikits.learn and was initially developed by David Cournapeau as a Google summer of code project in 2007. Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation), took this project at another level and made the first public release (v0.1 beta) on 1st Feb. 2010.

### **Version History:** -

- April 2021: scikit-learn 0.24.2
- Jan 2021: scikit-learn 0.24.1

- Dec 2020: scikit-learn 0.24.0
- May 2020: scikit-learn 0.23.2
- May 2020: scikit-learn 0.23.1
- May 2020: scikit-learn 0.23.0
- Jan 2020: scikit-learn 0.22.1
- Dec 2019: scikit-learn 0.22.0
- May 2019: scikit-learn 0.21.0
- March 2019: scikit-learn 0.20.3
- December 2018: scikit-learn 0.20.2
- November 2018: scikit-learn 0.20.1
- September 2018: scikit-learn 0.20.0
- July 2018: scikit-learn 0.19.2
- July 2017: scikit-learn 0.19.0
- September 2016: scikit-learn 0.18.0
- November 2015: scikit-learn 0.17.0
- March 2015: scikit-learn 0.16.0
- July 2014: scikit-learn 0.15.0
- August 2013: scikit-learn 0.14

Scikit Learn is built on top of several common data and math Python libraries. Such a design makes it super easy to integrate between them all. You can pass numpy arrays and pandas data frames directly to the ML algorithms of Scikit! It uses the following libraries:

NumPy: For any work with matrices, especially math operations

SciPy: Scientific and technical computing

Matplotlib: Data visualization

IPython: Interactive console for Python

Sympy: Symbolic mathematics

Pandas: Data handling, manipulation, and analysis

Scikit Learn is focused on Machine Learning, e.g data modelling. It is not concerned with the loading, handling, manipulating, and visualizing of data. Thus, it is natural and common practice to use the above libraries, especially NumPy, for those extra steps; they are made for each other!

Scikit's robust set of algorithm offerings include:

- Regression: Fitting linear and non-linear models
- Clustering: Unsupervised classification
- Decision Trees: Tree induction and pruning for both classification and regression tasks
- Neural Networks: End-to-end training for both classification and regression. Layers can be easily defined in a tuple
- SVMs: for learning decision boundaries

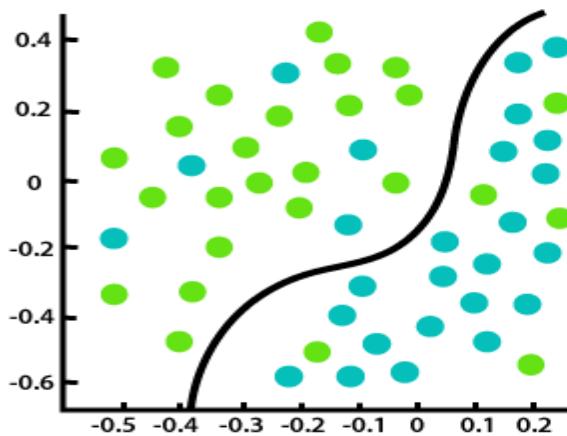
- Naive Bayes: Direct probabilistic modelling
- Even beyond that, it has some very convenient and advanced functions not commonly offered by other libraries:
- Ensemble Methods: Boosting, Bagging, Random Forest, Model voting and averaging
- Feature Manipulation: Dimensionality reduction, feature selection, feature analysis
- Outlier Detection: For detecting outliers and rejecting noise
- Model selection and validation: Cross-validation, Hyperparameter tuning, and metrics.

Now we have discussed about the machine learning library `sklearn` , let us start with the basic algorithms in machine learning.

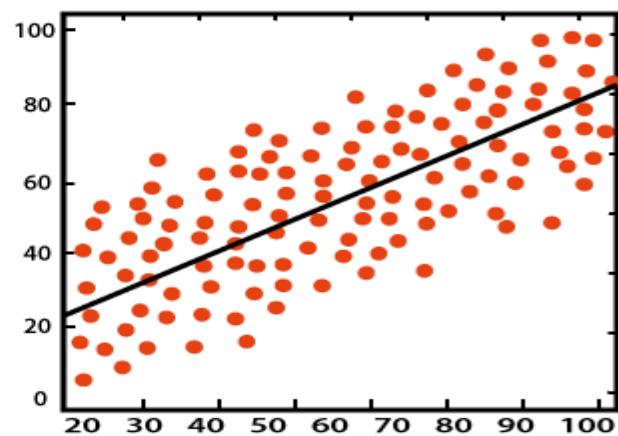
## 4.4 Regression vs Classification

Regression and Classification algorithms are Supervised Learning algorithms. Both the algorithms are used for prediction in Machine learning and work with the labelled datasets. But the difference between both is how they are used for different machine learning problems.

The main difference between Regression and Classification algorithms is that Regression algorithms are used to predict the continuous values such as price, salary, age, etc. whereas Classification algorithms are used to predict/Classify the discrete values such as Male or Female, True or False, Spam or Not Spam, etc.



Classification



Regression

Image: Classification vs Regression

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

## Classification

Classification is a process of finding a function which helps in dividing the dataset into classes based on different parameters. In Classification, a computer program is trained on the training dataset and based on that training, it categorizes the data into different classes.

The task of the classification algorithm is to find the mapping function to map the input (x) to the discrete output (y).

Example: The best example to understand the Classification problem is Email Spam Detection. The model is trained on the basis of millions of emails on different parameters, and whenever it receives a new email, it identifies whether the email is spam or not. If the email is spam, then it is moved to the Spam folder.

Types of ML Classification Algorithms:

Classification Algorithms can be further divided into the following types:

- i) Logistic Regression
- ii) K-Nearest Neighbours
- iii) Support Vector Machines
- iv) Kernel SVM
- v) Naïve Bayes
- vi) Decision Tree Classification
- vii) Random Forest Classification

## 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.

Types of Regression Algorithm:

- i) Simple Linear Regression
- ii) Multiple Linear Regression
- iii) Polynomial Regression
- iv) Support Vector Regression

- v) Decision Tree Regression
- vi) 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.

## 4.5 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)	# 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.

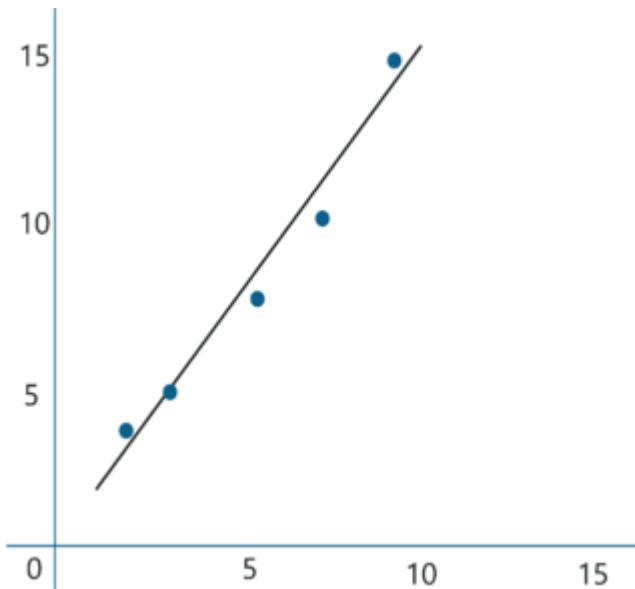
**Step 3:** Substitute the values in the final equation

$$y = mx + c$$

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

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).

## 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.

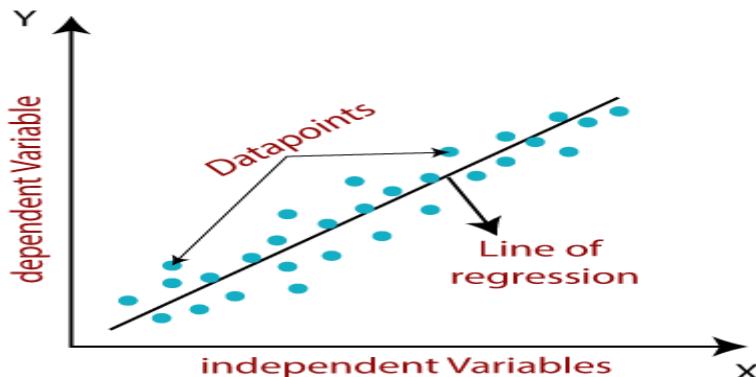


Image: Relationship between the variables in linear regression

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).

$\epsilon$  = 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:

### 1. 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.

### 2. Multiple Linear regression:

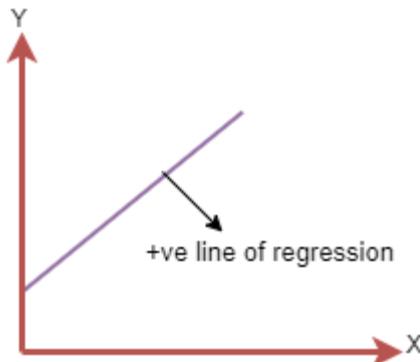
If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

## 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:

- 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.



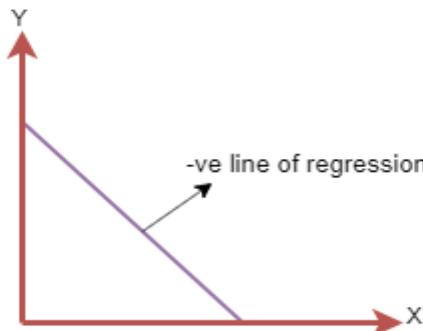
The line equation will be:  $Y = a_0 + a_1 x$

Image:Positive linear relationship

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

- 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:  $Y = -a_0 + a_1x$

Image:Negative linear relationship

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

## 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

$(a_1x_i + a_0)$  = 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 be 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.

### Model Performance:

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:

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

### 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.

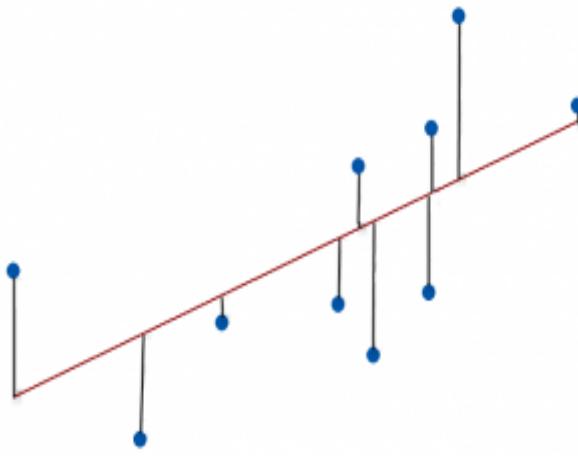


Image: Ordinary least squares line

Reference: <https://statisticsbyjim.com/glossary/ordinary-least-squares/#:~:text=Ordinary%20least%20squares%2C%20or%20linear,and%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.

## Project : Linear Regression in Sklearn

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.

Link for Project :

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/1.Linear%20Regression.ipynb>

We have discussed about regression, ordinary least square method, gradient descent optimization technique and metrics to measure the performance of regression. Now, let's get started with the cloud platform where we can train, test and deploy our machine learning model.

## 4.6 Azure ML No Code Platform

Machine Learning is the foundation for most artificial intelligence solutions, and the creation of an intelligent solution often begins with the use of machine learning to train a predictive model using historic data that you have collected. Azure Machine Learning is a cloud service that you can use to train and manage machine learning models.

Azure Machine Learning is a cloud service that allows building no-code machine learning models through a drag and drop visual interface. It's designed to help data scientists and machine learning engineers to leverage their existing data processing and model development skills & frameworks.

Set of Azure  
Cloud Services



Python  
SDK

That enables  
you to:

- ✓ Prepare Data
- ✓ Build Models
- ✓ Train Models

- ✓ Manage Models
- ✓ Track Experiments
- ✓ Deploy Models

Image: Azure Machine Learning

Reference: [https://docs.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-ml?WT.mc\\_id=aiml-0000-abornst](https://docs.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-ml?WT.mc_id=aiml-0000-abornst)

## Azure ML Studio Briefing

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.

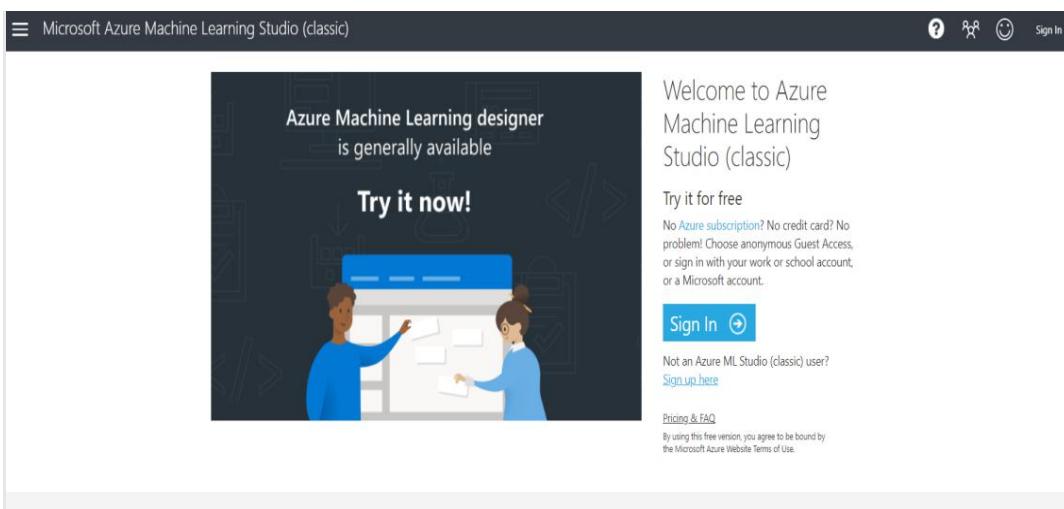
- The following steps are the most important steps in the entire Azure machine learning process. The module “Train Model” accepts two input parameters. First is the raw training data, and the other is the learning algorithm.
- Evaluate Model module gives us an evaluation of the trained model expressed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

## PREDICTING NEW DATA

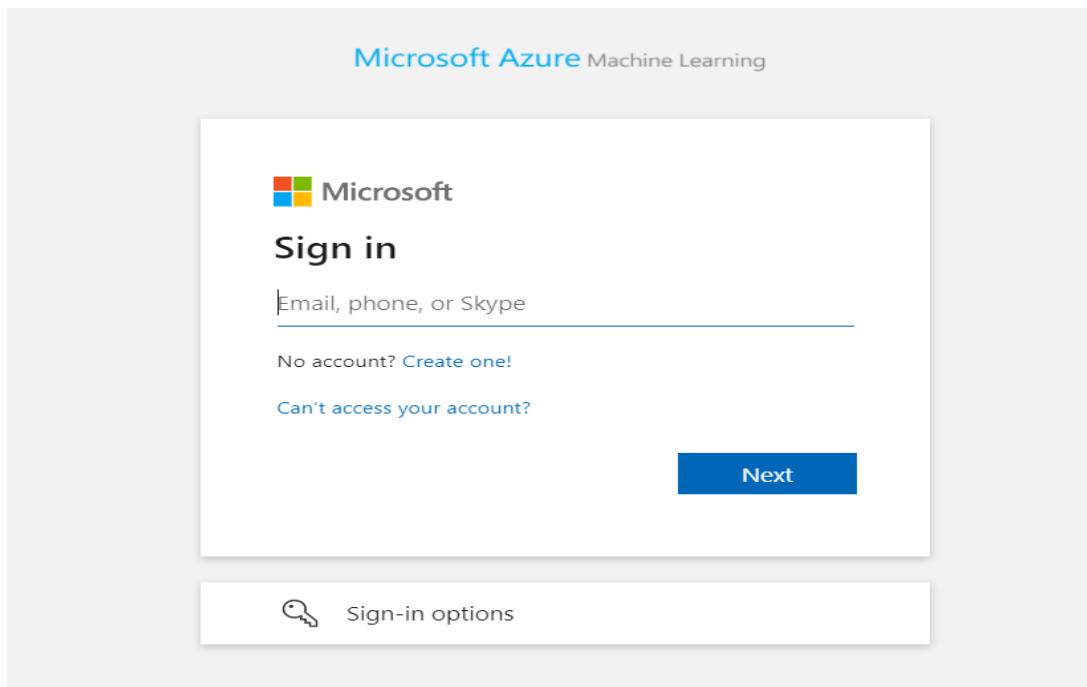
Finally, we can test our prediction web service using a simple test form.

## 4.7 Setting up an account in Azure ML Studio and Creating Workspace

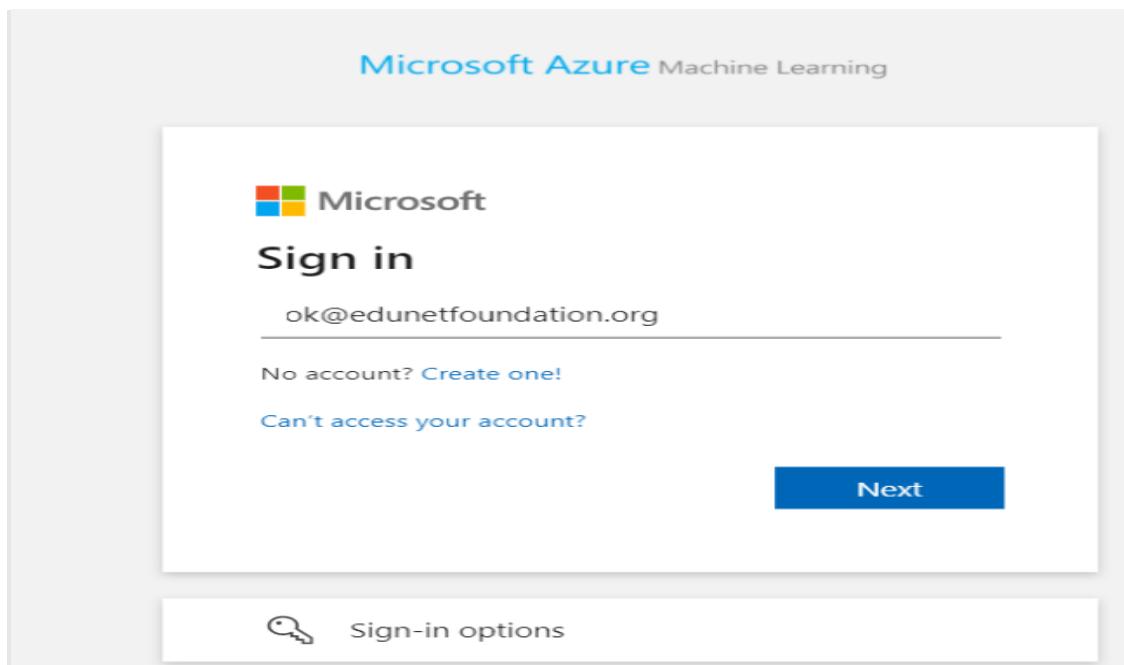
- Open the following web link on the web browser.  
<https://studio.azureml.net/>



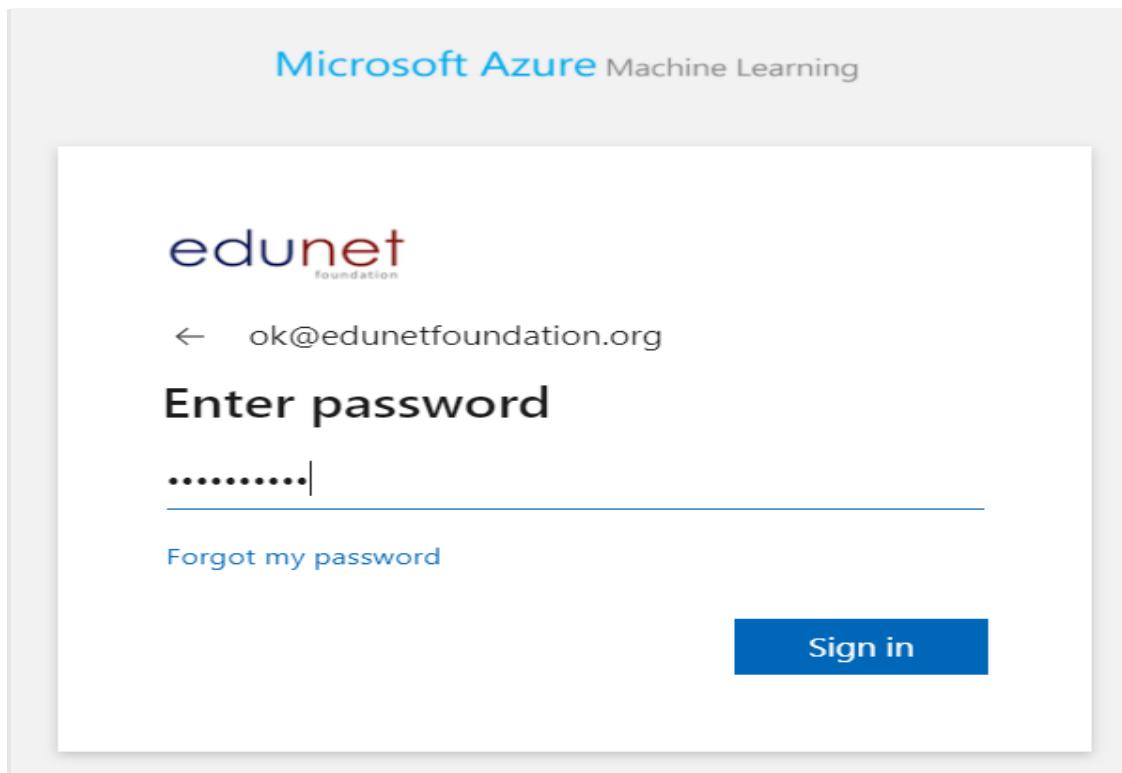
- Click on the Sign In.



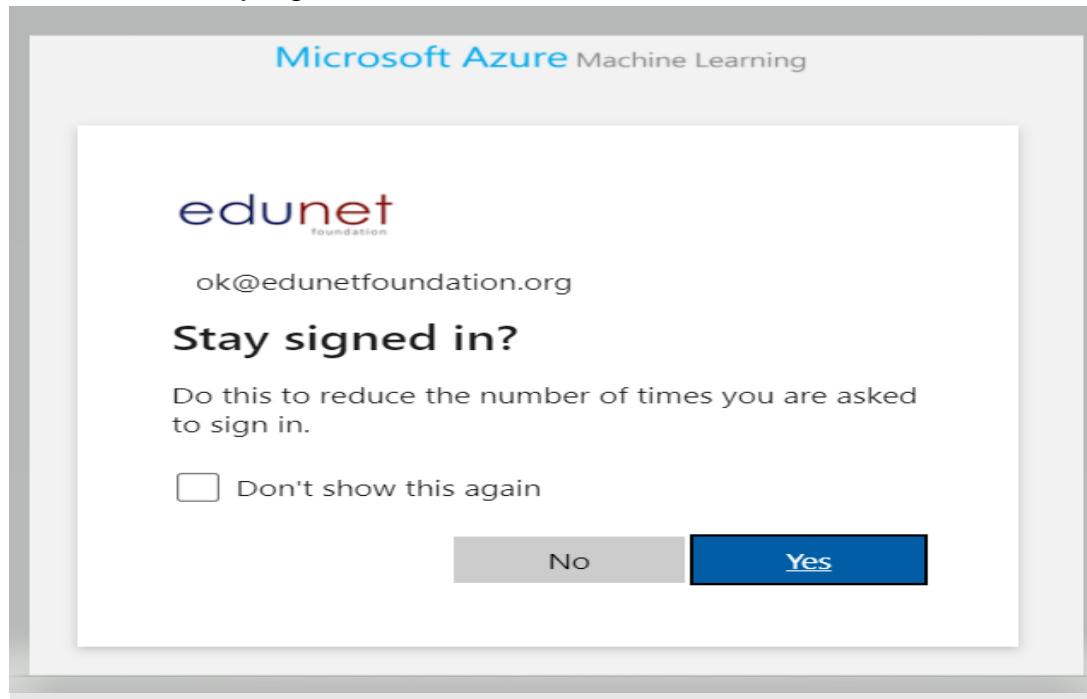
- Type your Microsoft email ID and then press next. If you do not have the Microsoft account then create first using **create one** option (<https://signup.live.com/>).



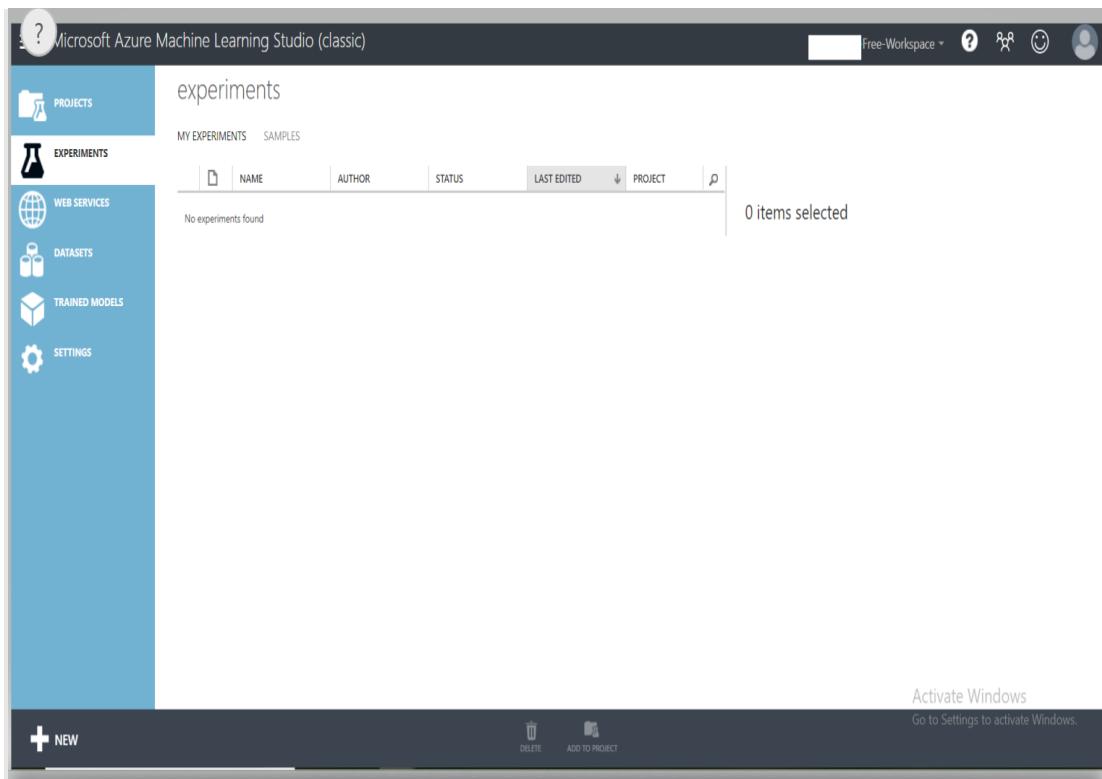
- Type the password for your Microsoft account and press sign in.



- Press Yes to stay sign in.

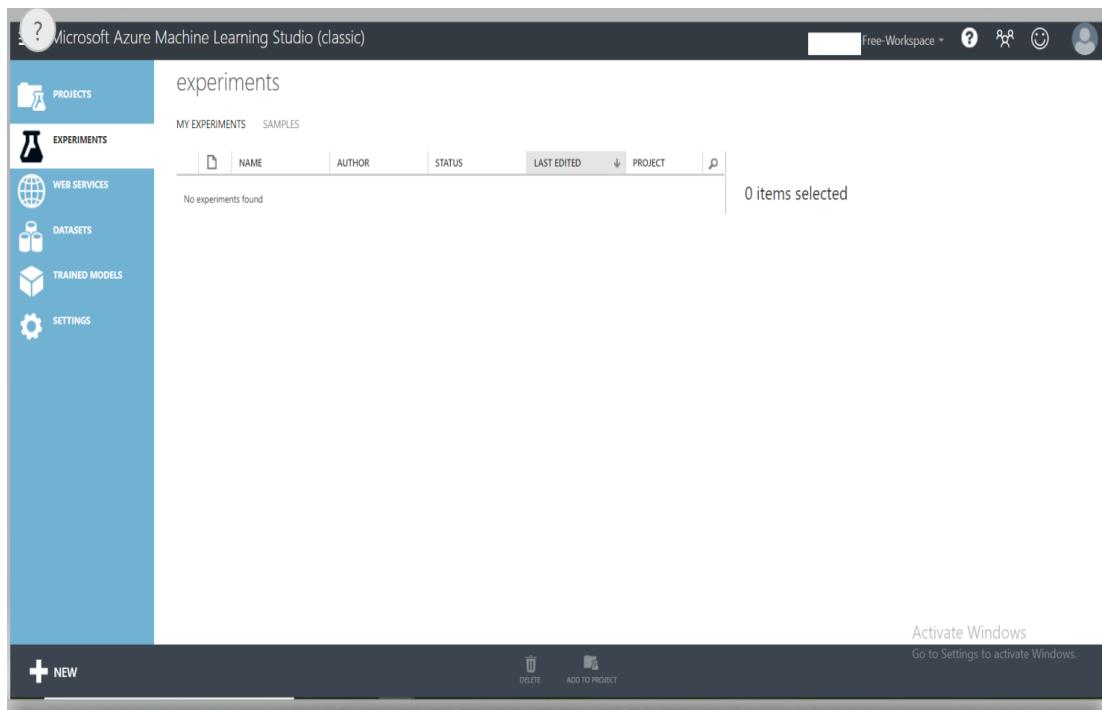


- Now, You will be redirected into following Microsoft Azure Machine Learning Studio (Classic) and your free workspace will be created as below:

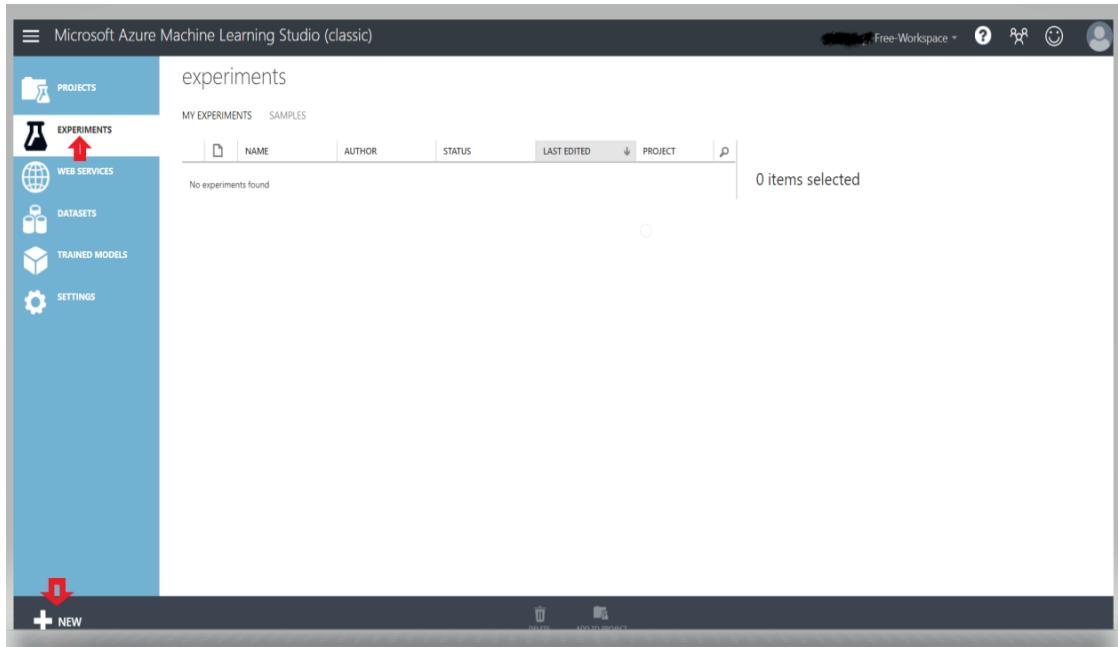


## 4.8 Training a ML model in Azure Studio

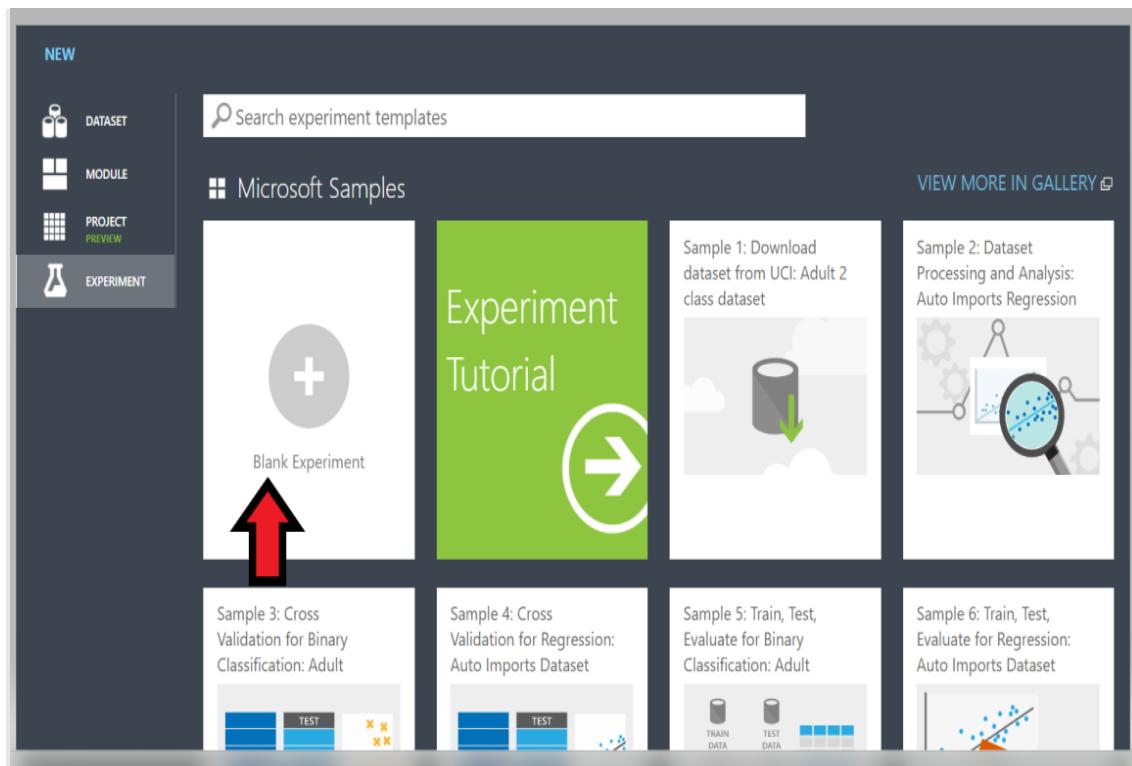
- Sign in into Microsoft Azure Machine Learning Studio (classic) and create workspace.



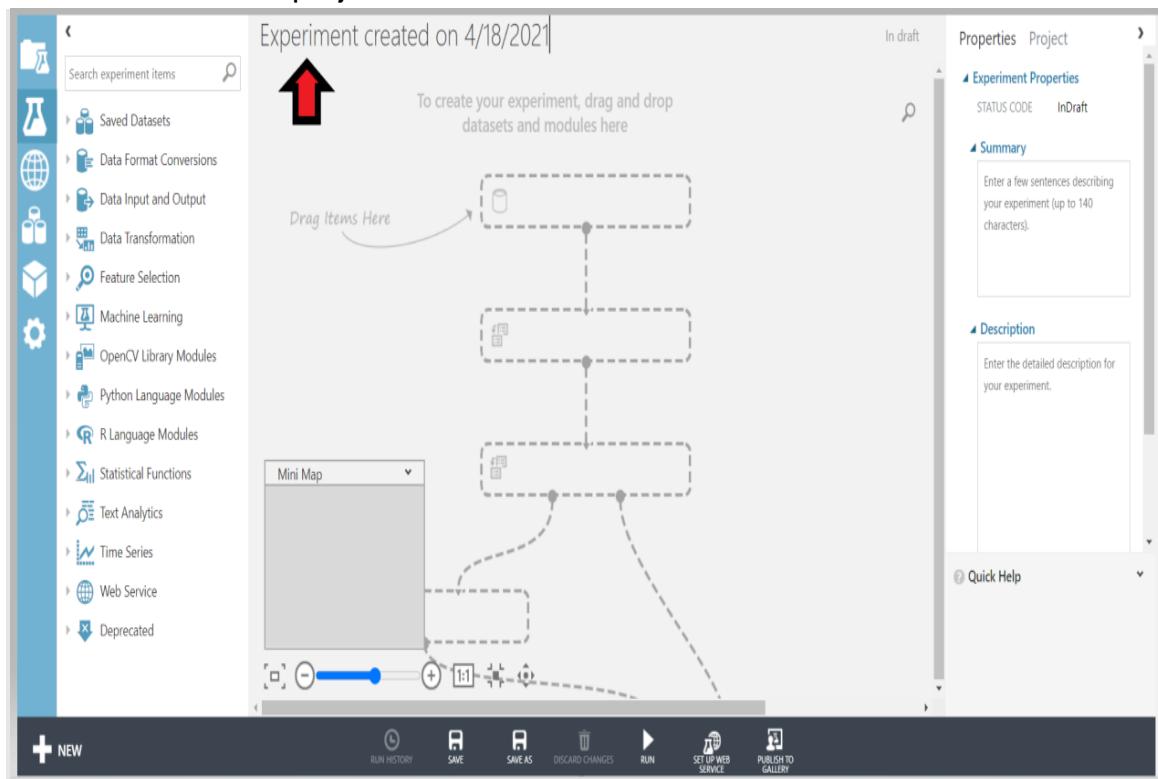
- First select **Experiment** and then **New** at the bottom of the page.



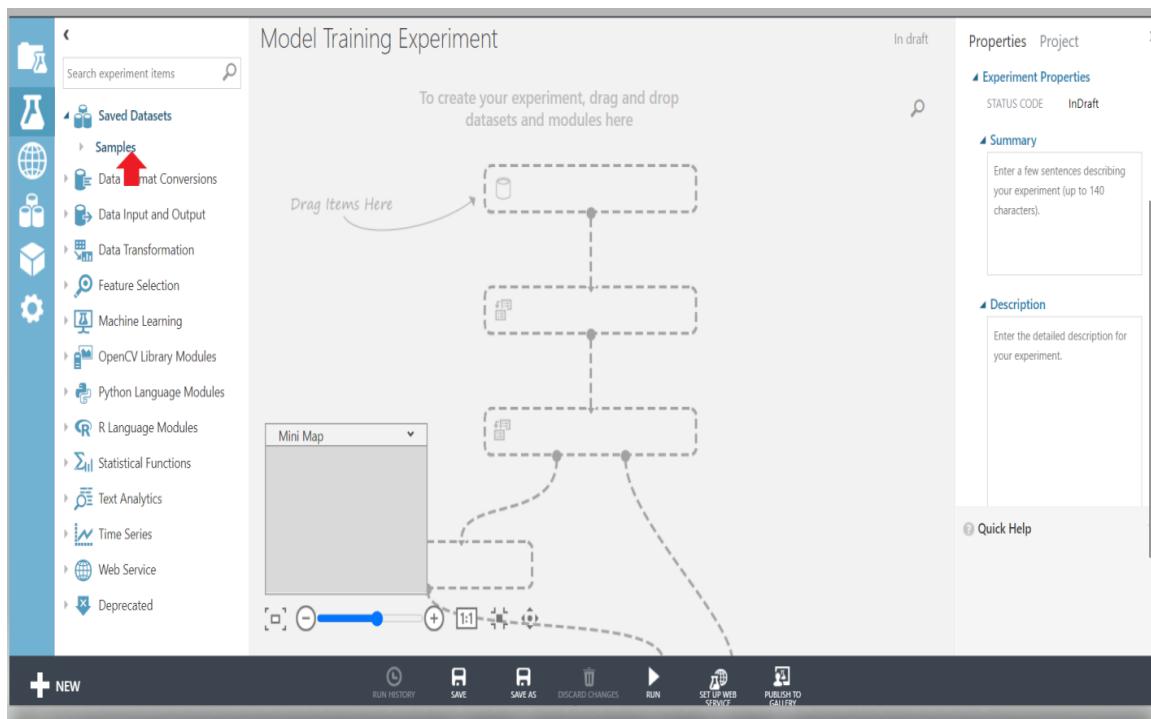
- Select Blank Experiment.



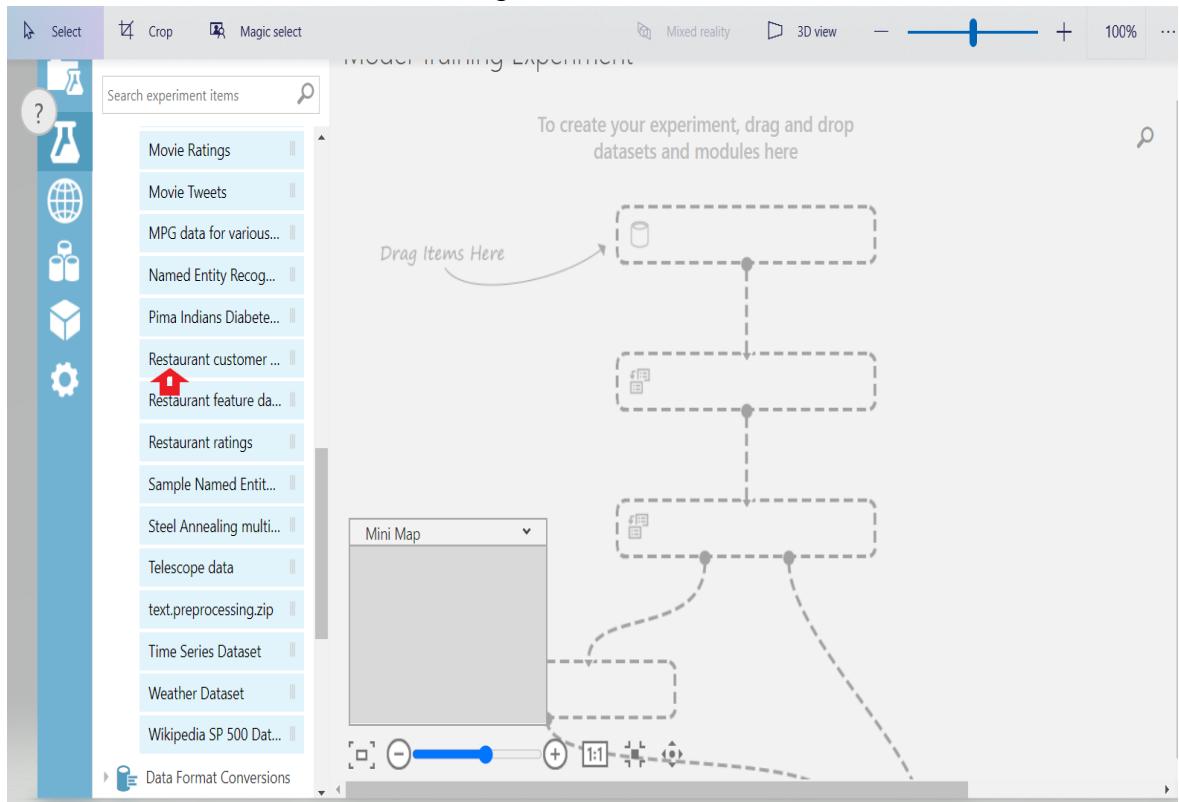
- Give the title for the project.



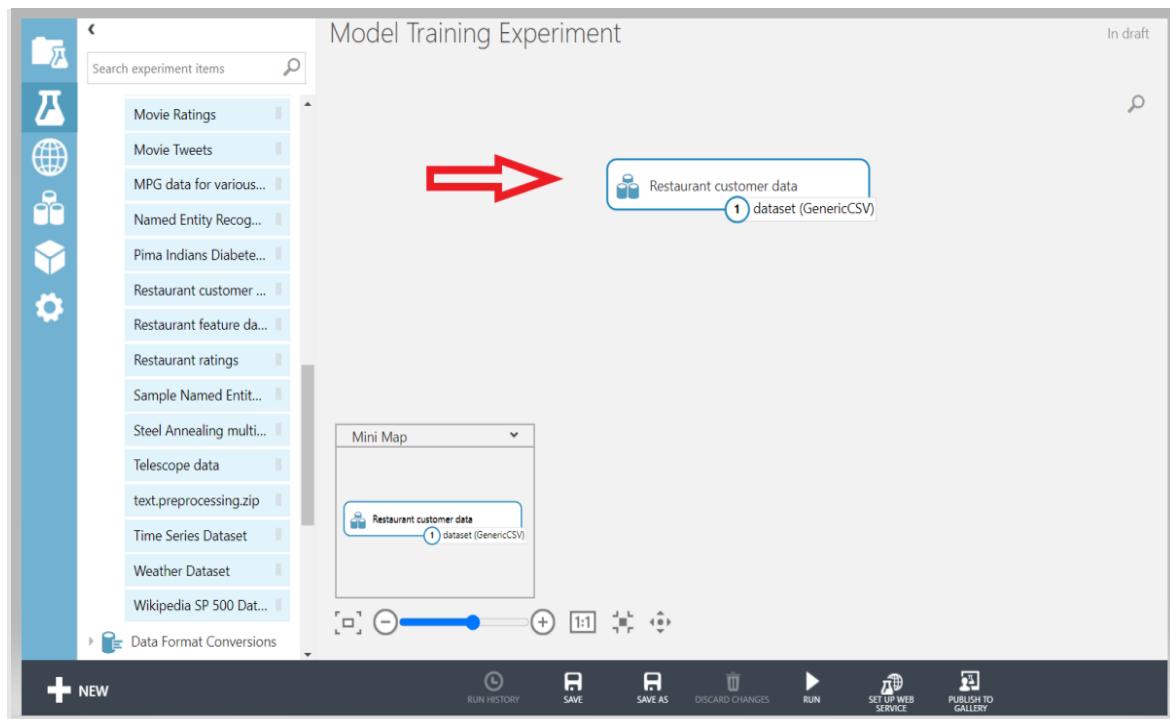
- Select **Sample** option from the **Saved Dataset**.



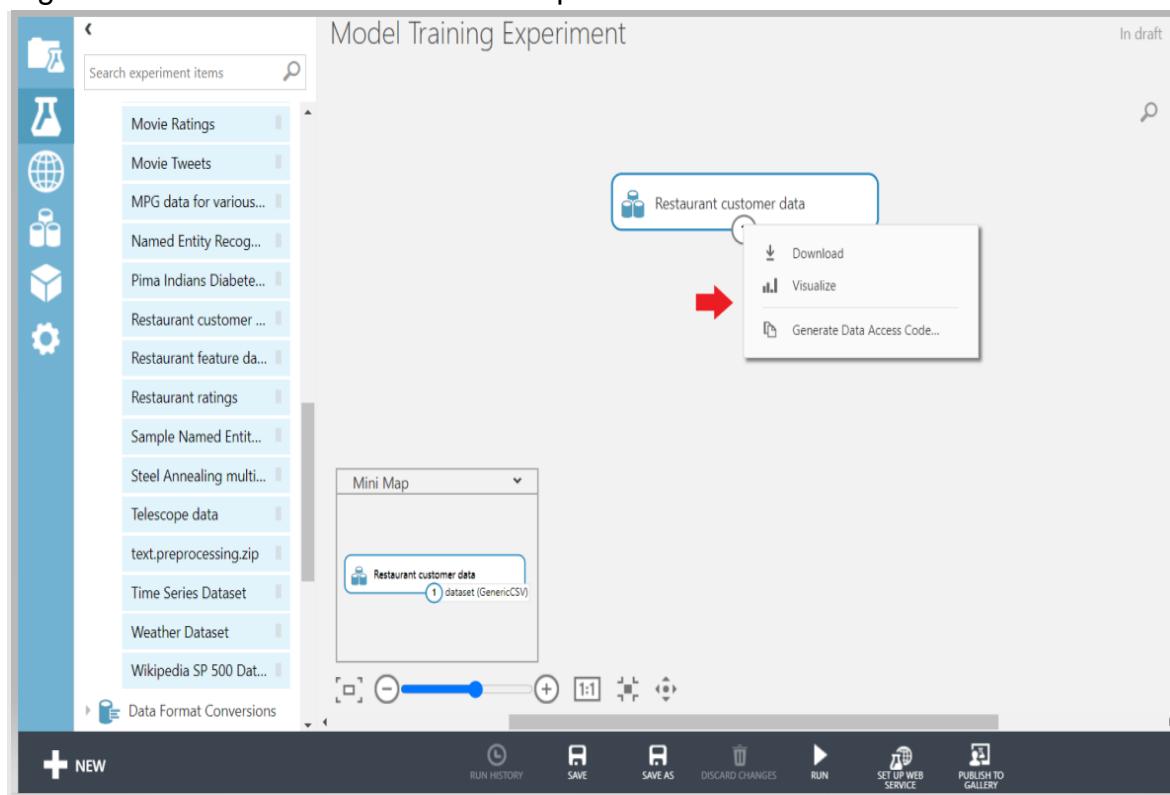
- Select Restaurant Customer Rating Dataset



- Drag selected dataset on Panel.



- Right click on 1 and choose visualize option.

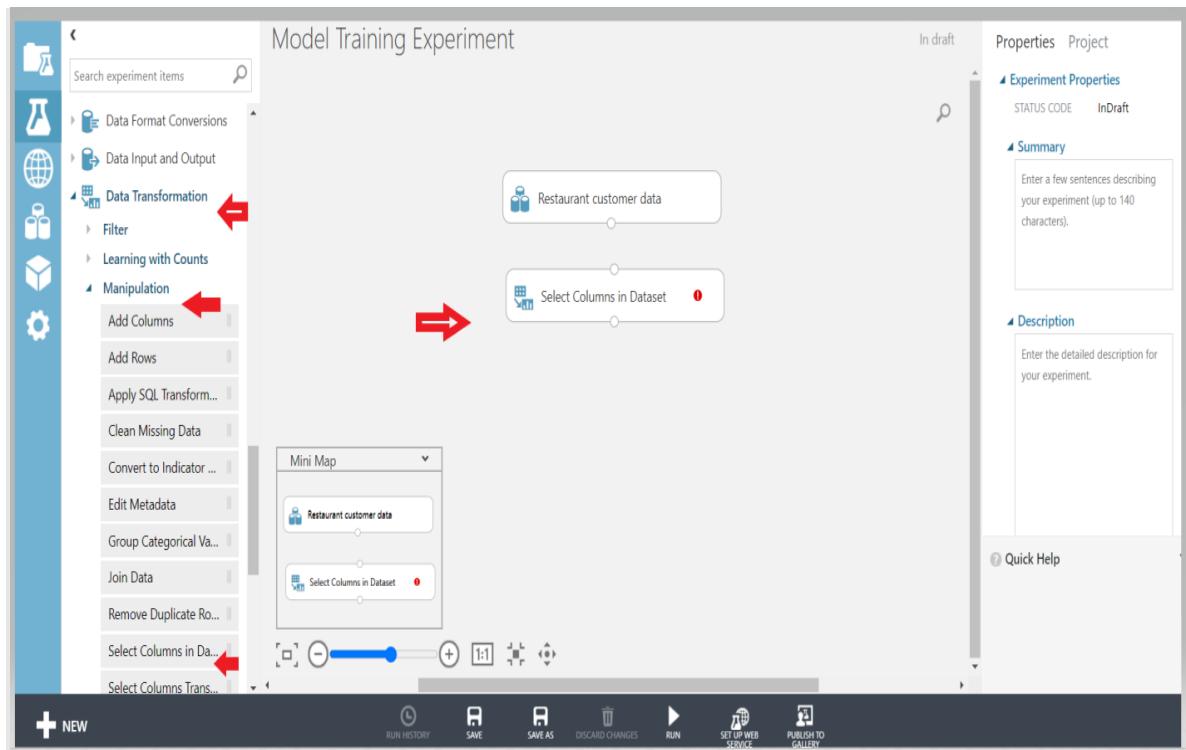


- Visualize the Dataset and then close it:

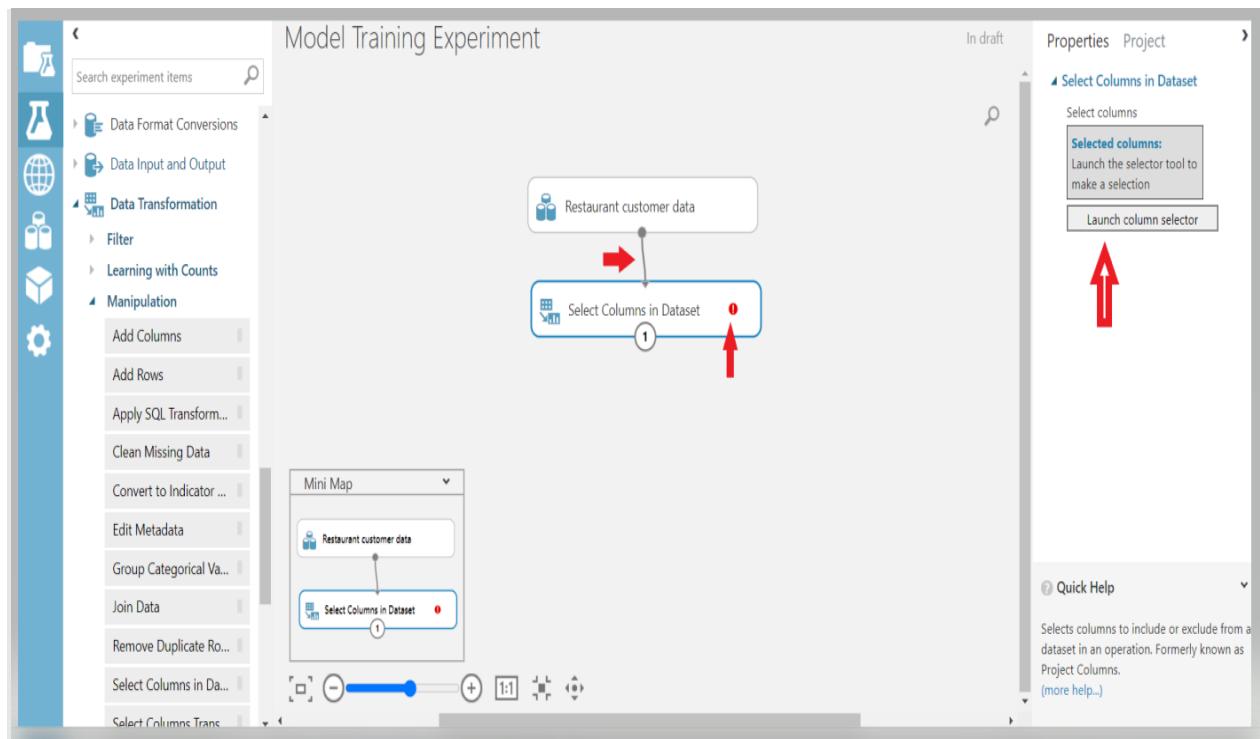
Model Training Experiment > Restaurant customer data > dataset

rows	columns
138	19
view as	
U1001	22.139997
U1002	22.150087
U1003	22.119847
U1004	18.867
U1005	22.183477
U1006	22.15
U1007	22.118464
U1008	22.122989
latitude	-100.978803
longitude	-100.983325
smoker	false
drink_level	abstemious
dress_preference	informal
ambience	family
transport	on foot

- Select the Data Transformation → Manipulation → Select Columns in Dataset and drag it into Panel.

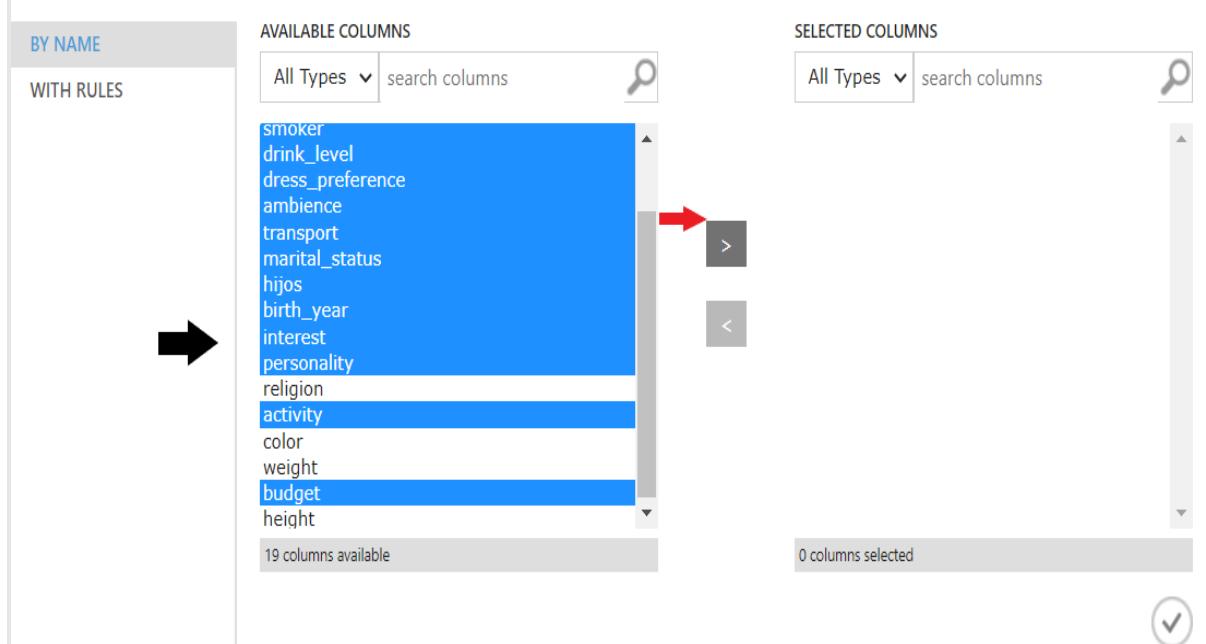


- Make connection between dragged item, press on red sign and then Launch column selector to choose the relevant column.



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

### Select columns



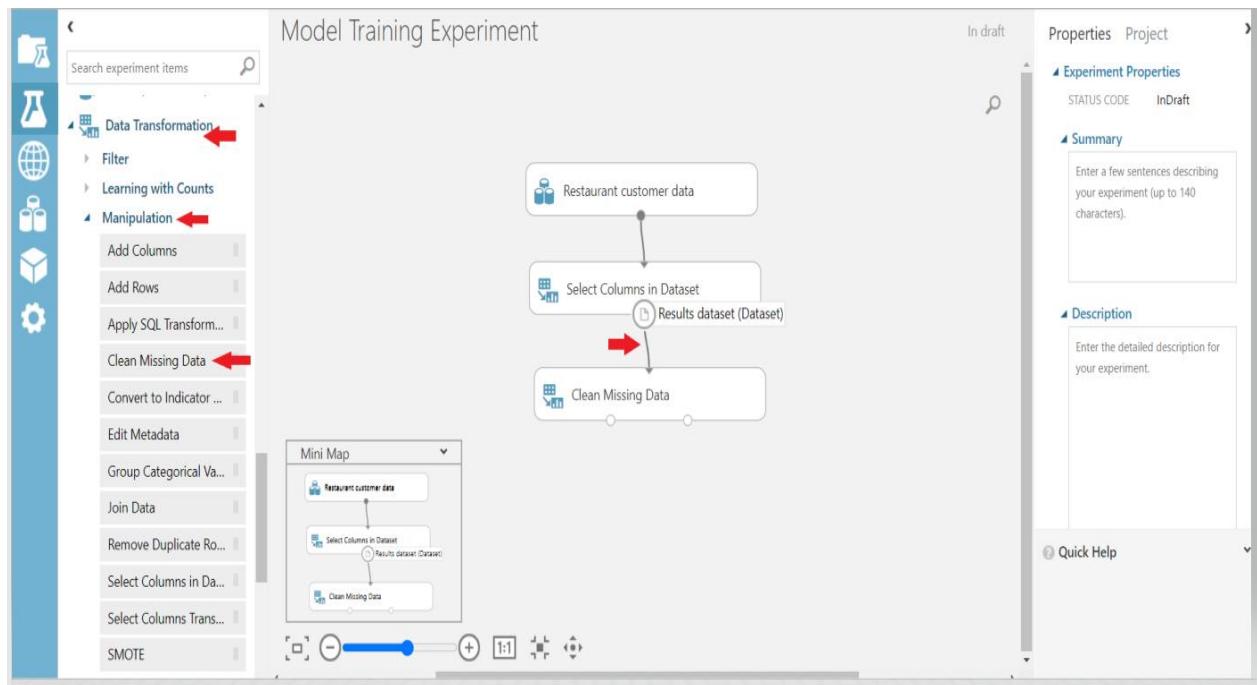
The screenshot shows the 'Select Columns in Dataset' dialog. The 'AVAILABLE COLUMNS' list on the left contains 19 columns: smoker, drink\_level, dress\_preference, ambience, transport, marital\_status, hijos, birth\_year, interest, personality, religion, activity, color, weight, budget, and height. The 'SELECTED COLUMNS' list on the right is empty. A large red arrow points from the 'WITH RULES' section of the main interface to this dialog. Another red arrow points to the right arrow button (>) in the column selector interface.

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

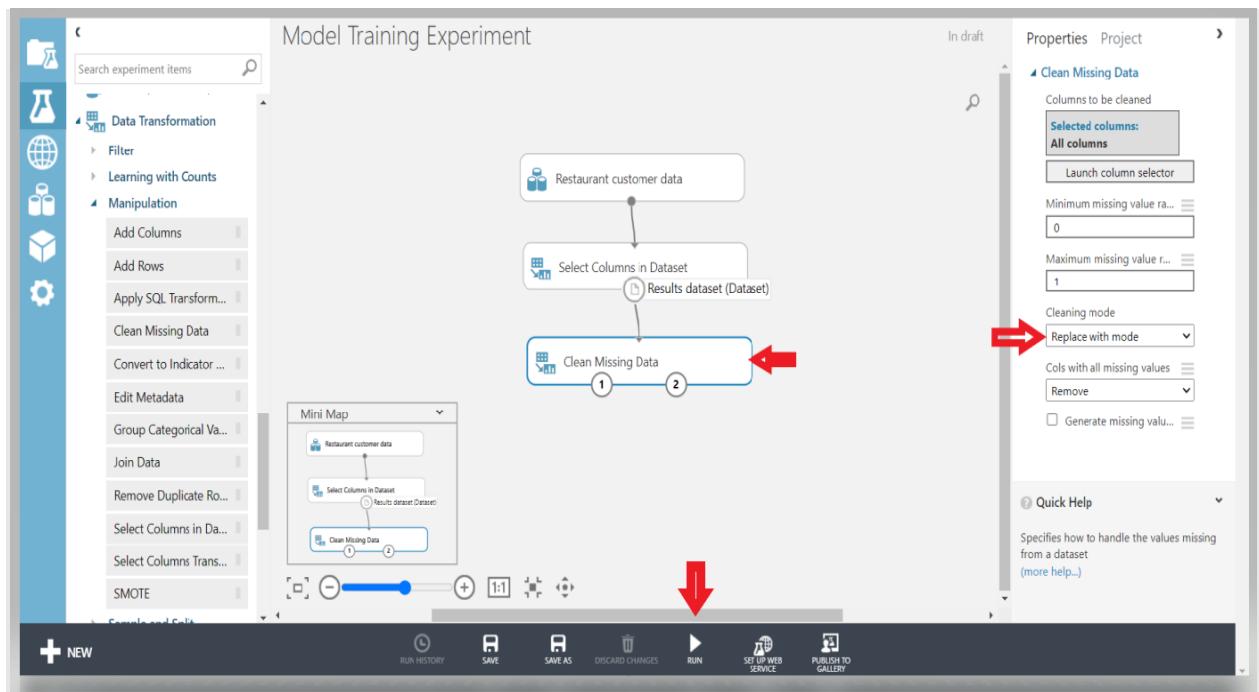
## Select columns

The screenshot shows the Power BI Data View interface. On the left, there's a sidebar with 'BY NAME' and 'WITH RULES' sections. The main area has two columns: 'AVAILABLE COLUMNS' and 'SELECTED COLUMNS'. In the 'AVAILABLE COLUMNS' section, there's a dropdown for 'All Types' and a search bar. Below it is a list of columns: userID, latitude, longitude, religion, color, weight, height. To the right of this list are two large grey buttons with arrows: one pointing right labeled '>', and one pointing left labeled '<'. At the bottom of this section, a grey bar says '7 columns available'. In the 'SELECTED COLUMNS' section, there's also a 'All Types' dropdown and a search bar. Below it is a list of columns: smoker, drink\_level, dress\_preference, ambience, transport, marital\_status, hijos, birth\_year, interest, personality, activity, budget. At the bottom of this section, a grey bar says '12 columns selected'. A red arrow points from the bottom right towards the bottom of the 'SELECTED COLUMNS' list.

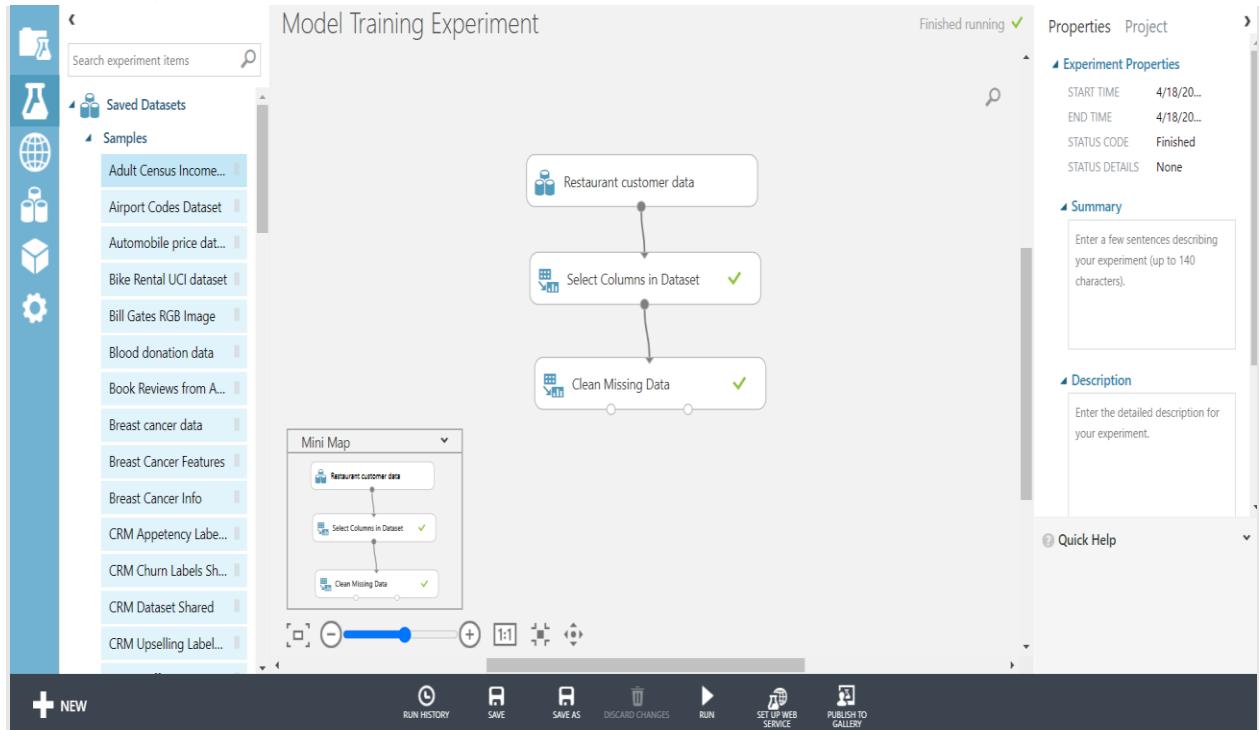
- Select the Data Transformation → Manipulation → Clean missing data, drag it into Panel and make connection.



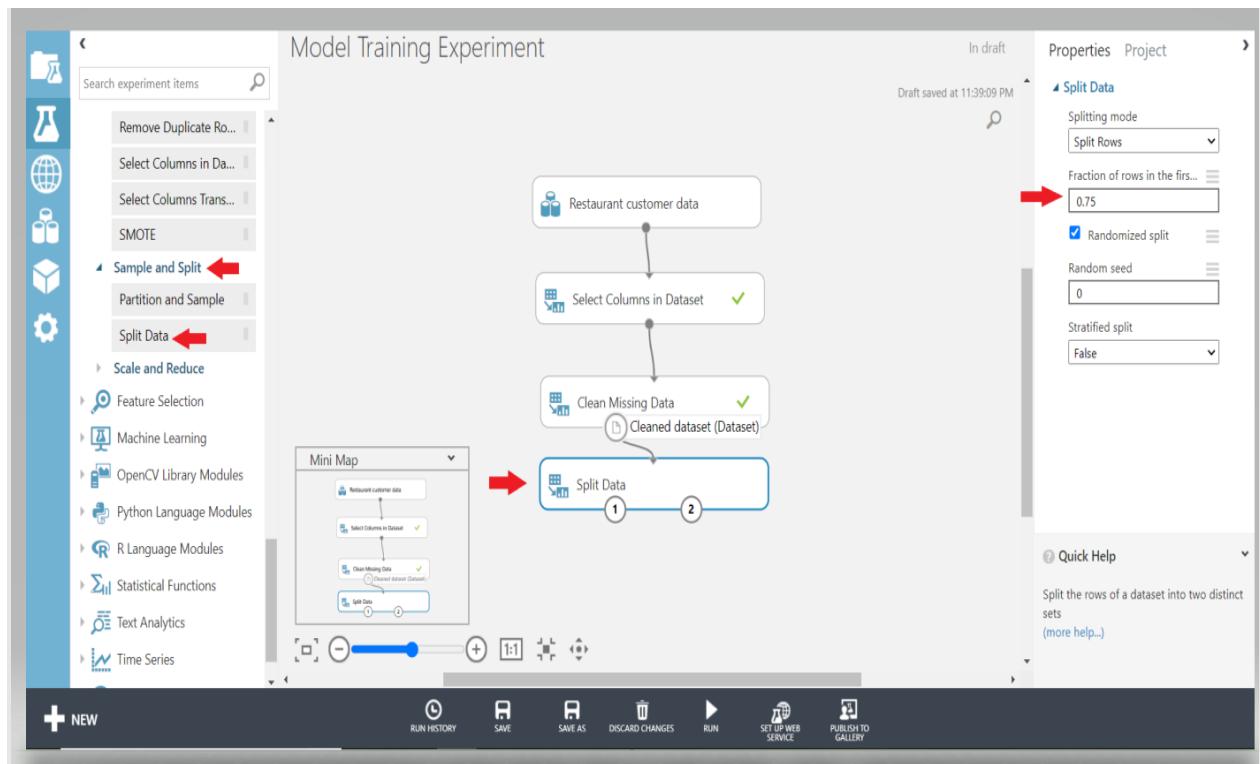
- Click on Clean missing data, set Replace with mode in cleaning mode and press Run.



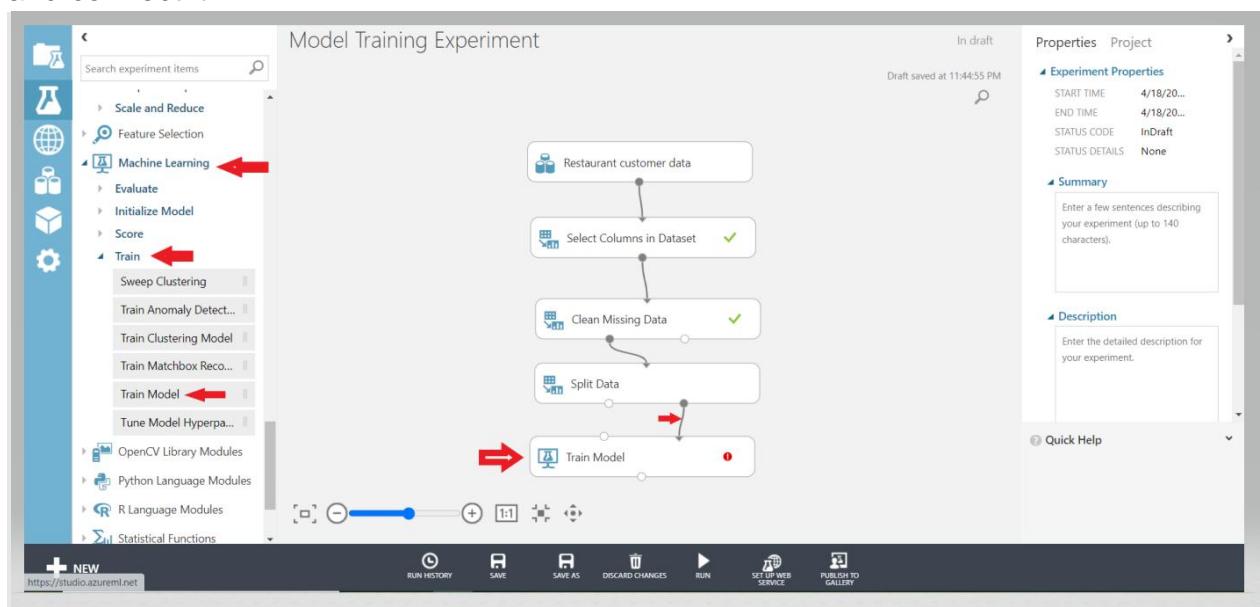
- If there is green tick, it means no error.



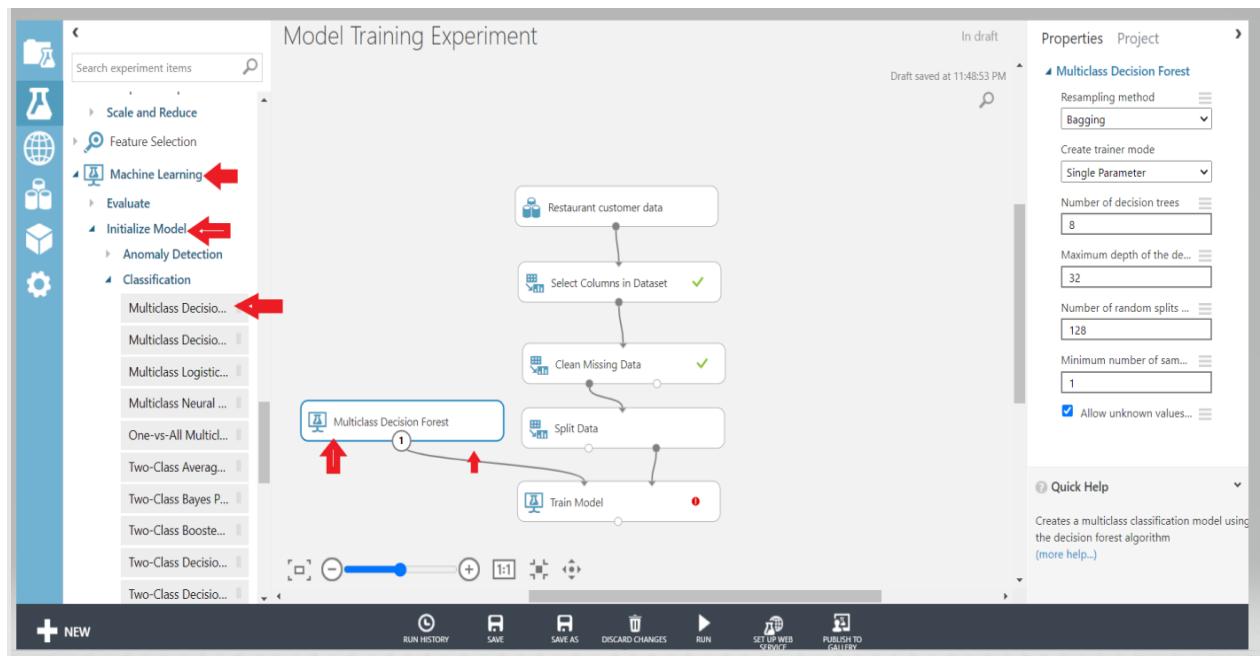
- Select Data Transformation → Sample and Split → Split data and drag it into Panel. Select the value 0.75 to split the data.



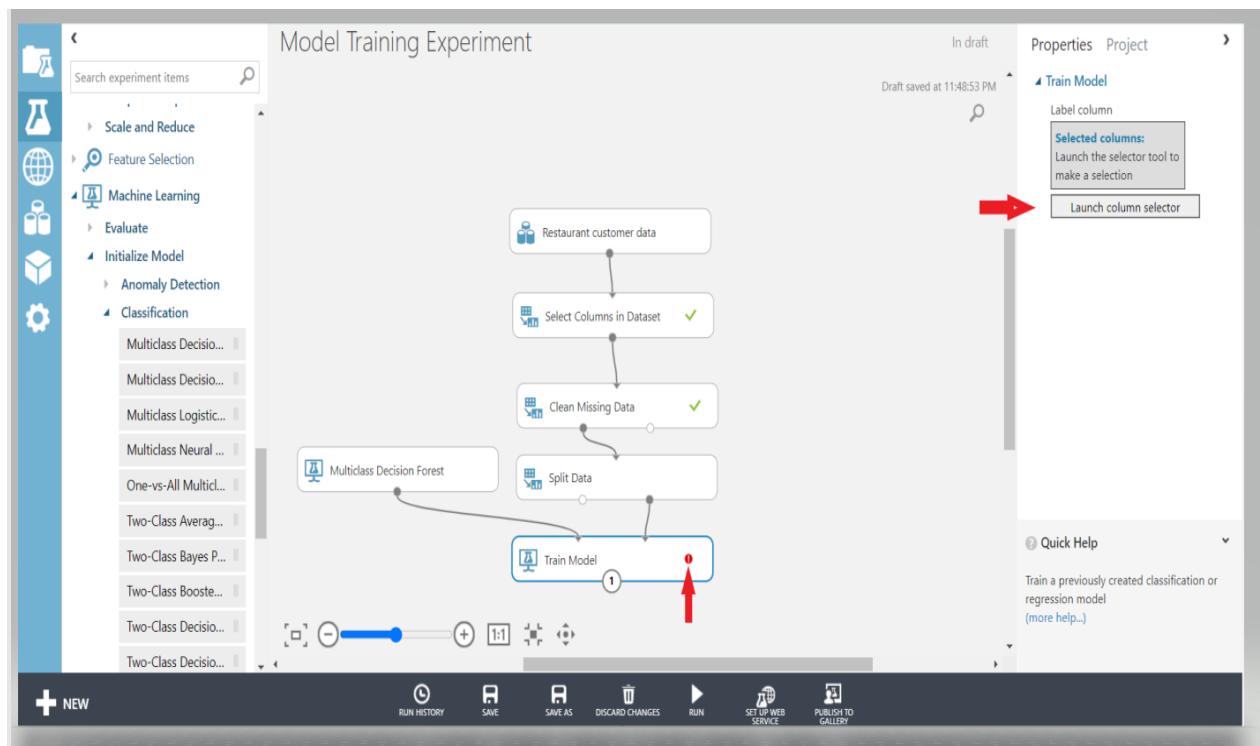
- Select Machine Learning → Train → Train Model and drag it into the Panel and connect it.



- Select the machine learning Algorithm from Machine Learning → Initialize Model → Classification → Multiclass Decision Forest, drag it into panel and make connection.

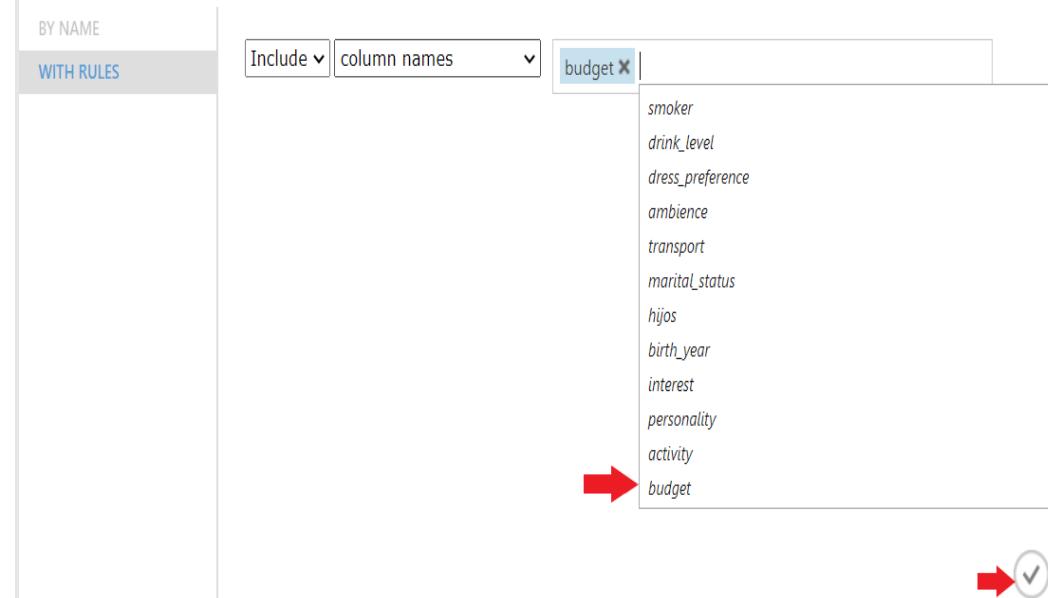


- Click on Train Model and Launch Column Selector



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

## Select a single column



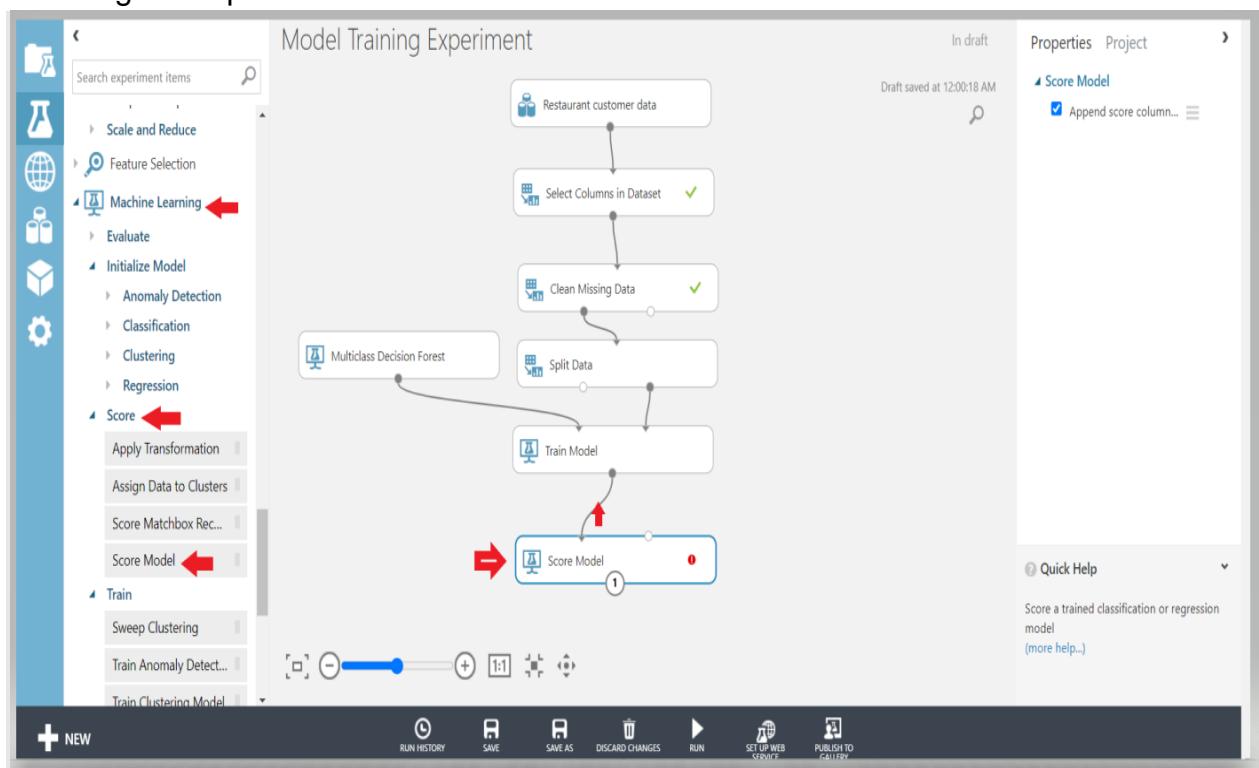
BY NAME

WITH RULES

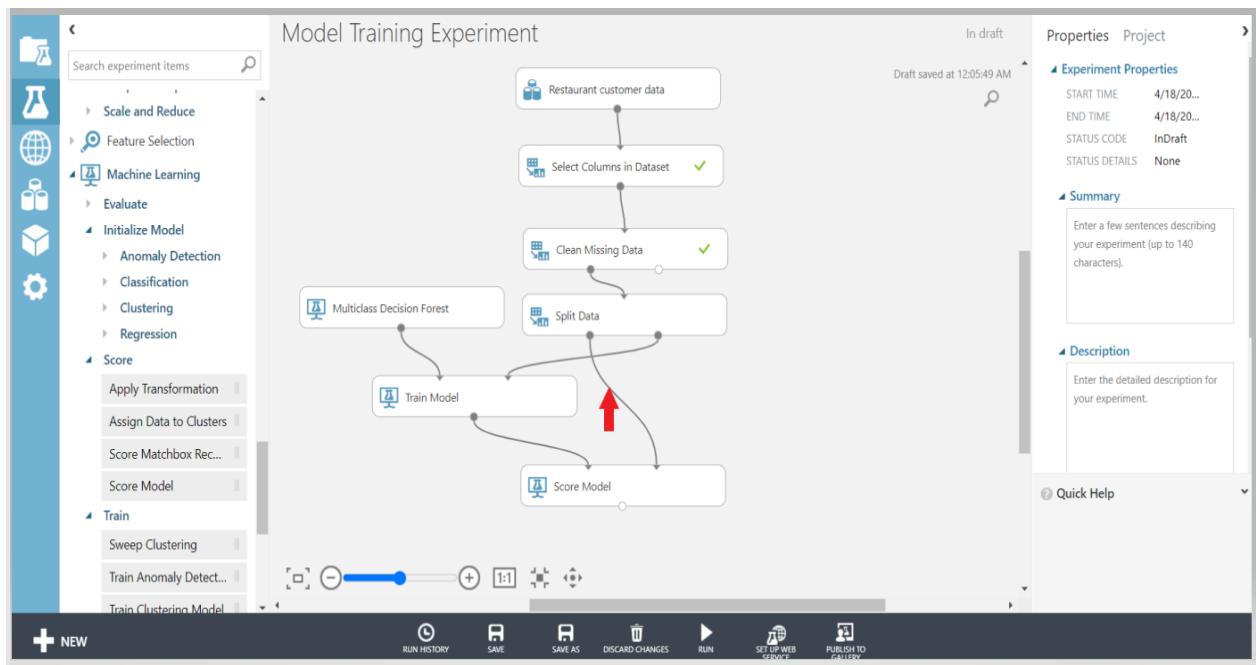
Include  column names

- budget
- smoker
- drink\_level
- dress\_preference
- ambience
- transport
- marital\_status
- hijos
- birth\_year
- interest
- personality
- activity
- budget

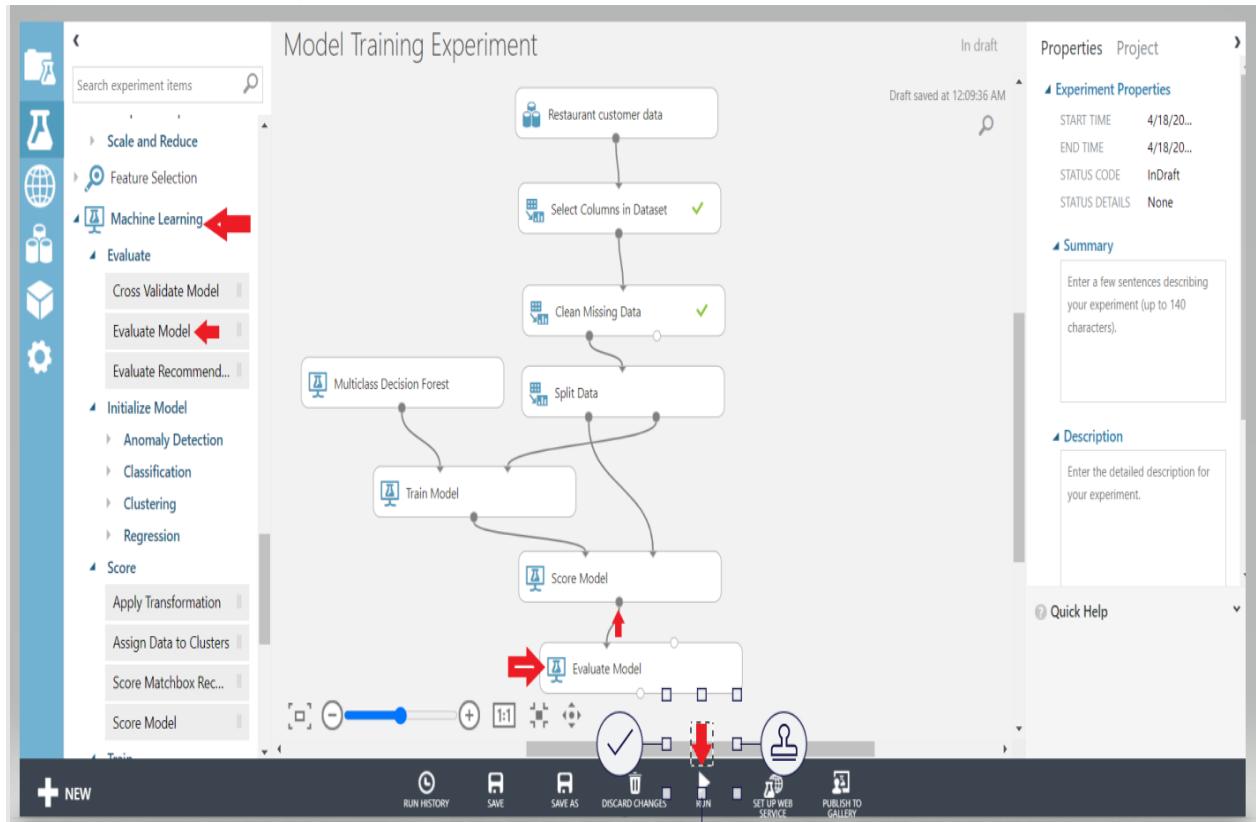
- For Score calculation, Choose Machine Learning → Score → Score Model and drag it into panel and Connect it.



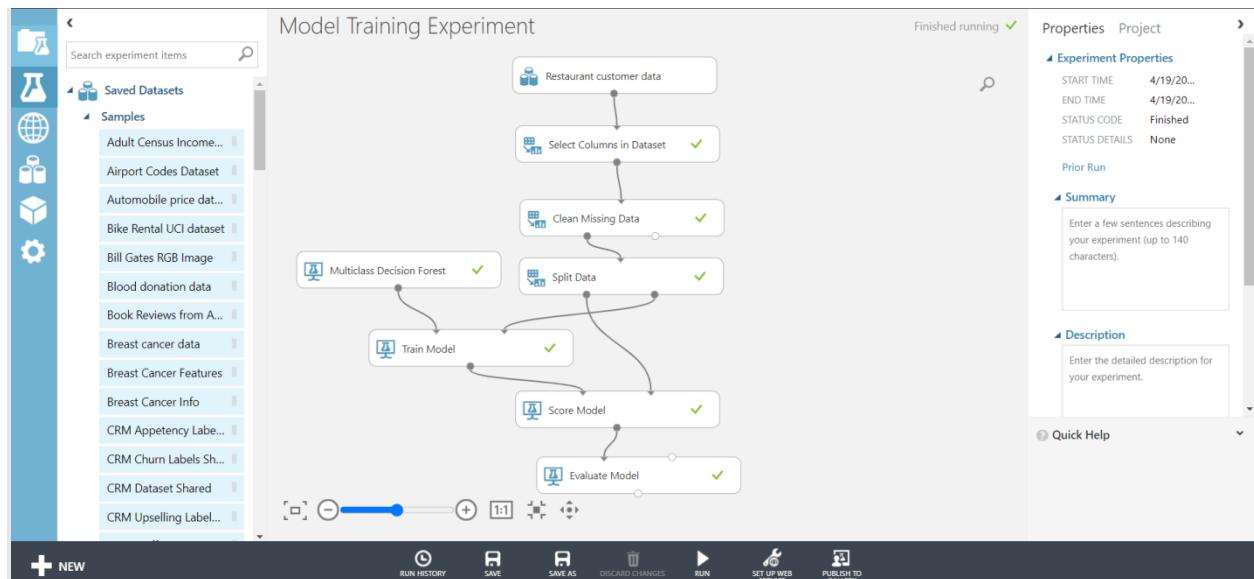
- Connect Split Data with Score Model for Testing.



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



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

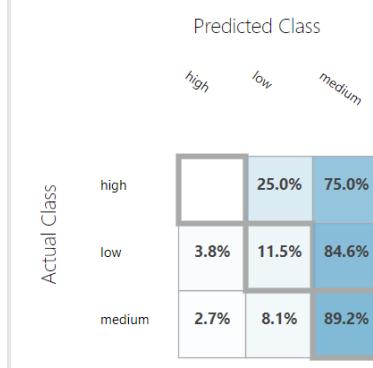


- To see the visualization Result, right click on Evaluate model → Evaluation Result → Visualize

Model Training Experiment > Evaluate Model > Evaluation results

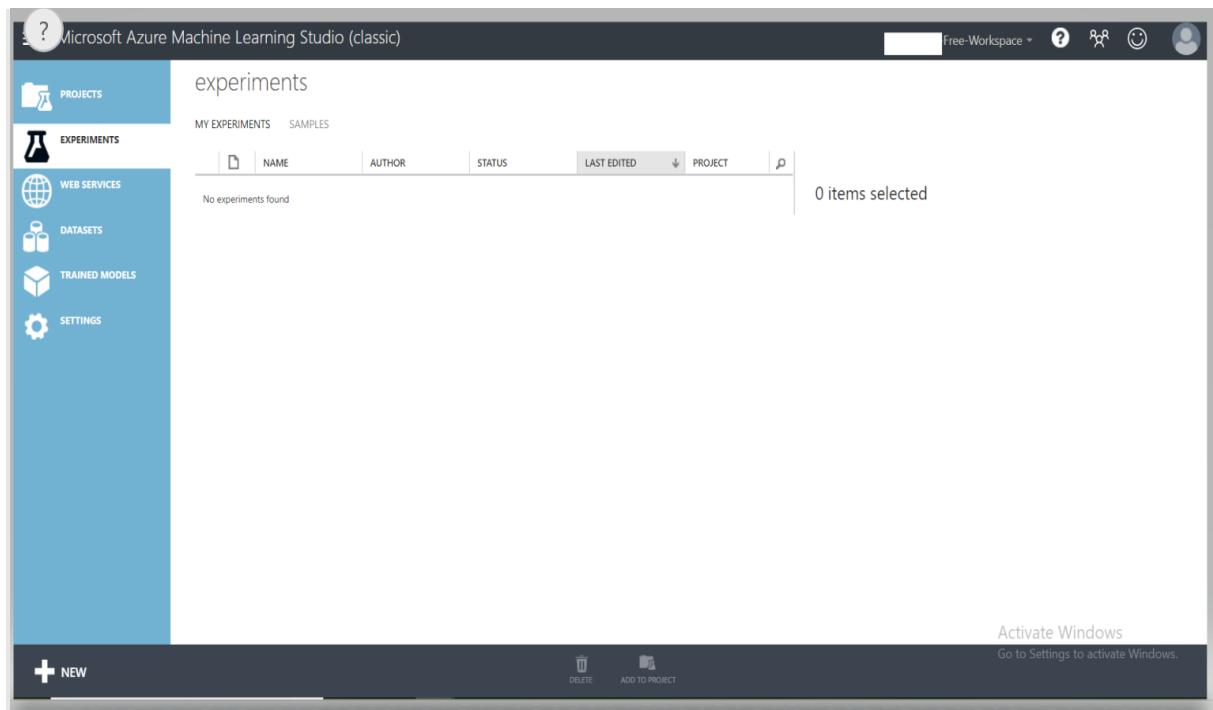
Average accuracy	0.775641
Micro-averaged precision	0.663462
Macro-averaged precision	0.341758
Micro-averaged recall	0.663462
Macro-averaged recall	0.335759

#### Confusion Matrix

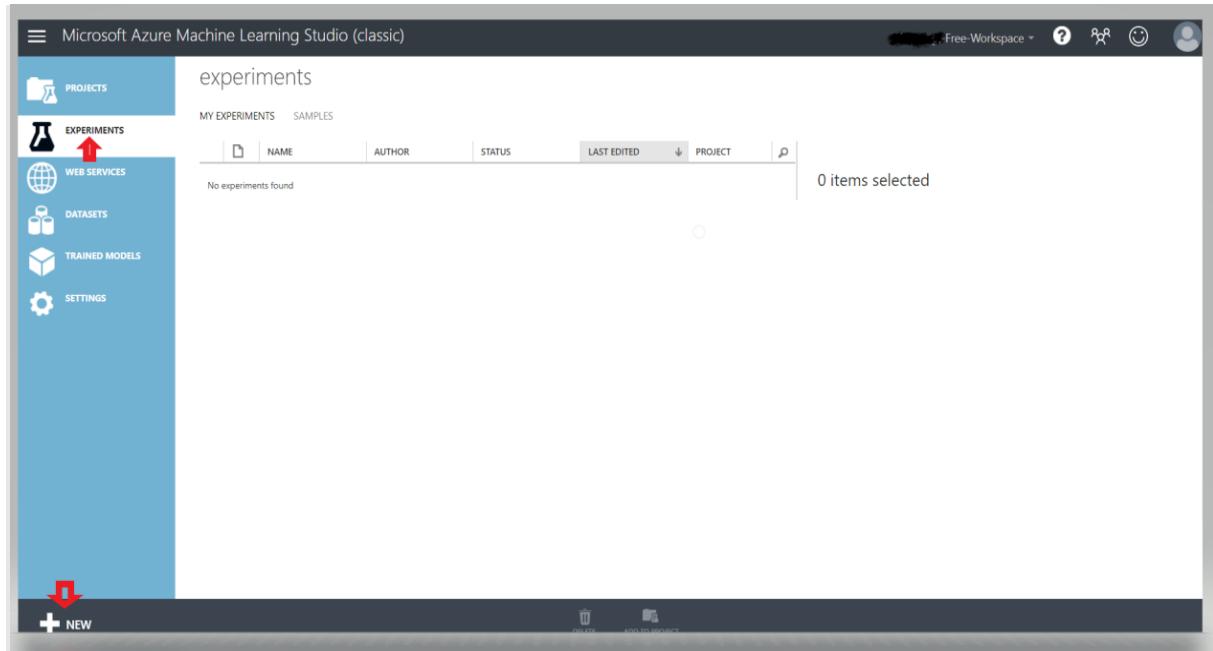


## 4.9 Create a Regression model with Azure ML Studio

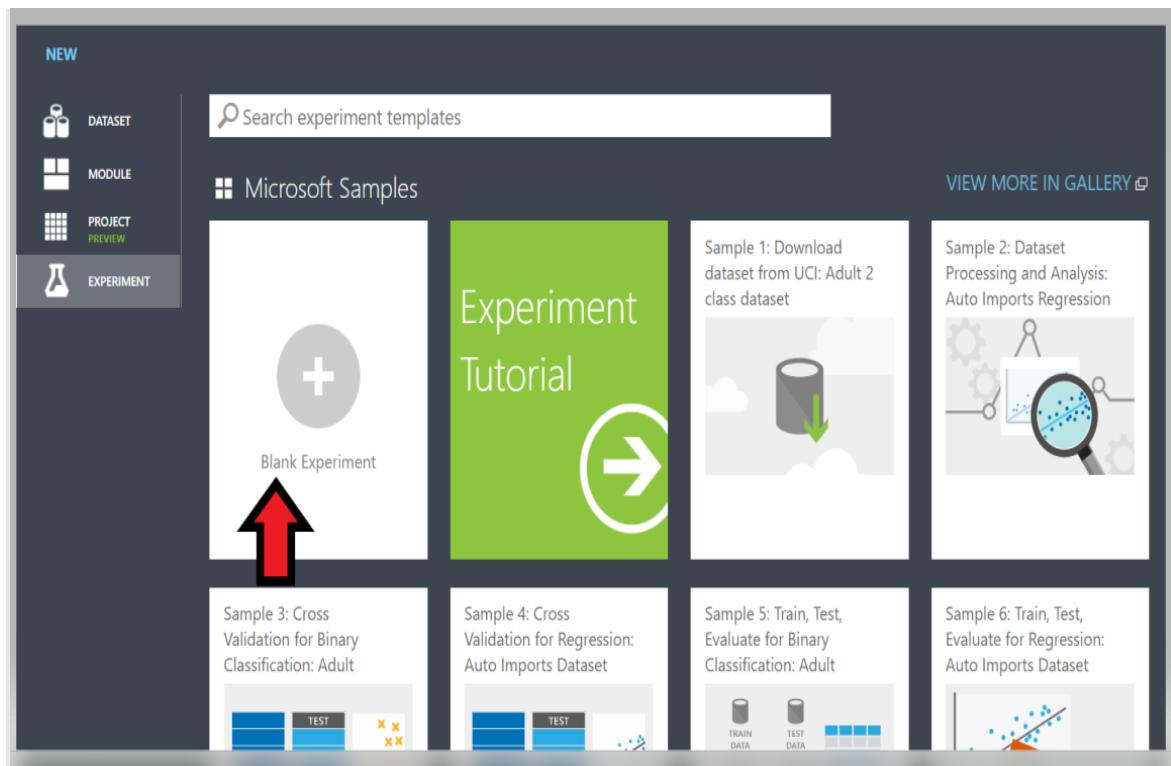
- Sign in into Microsoft Azure Machine Learning Studio (classic) and create workspace.



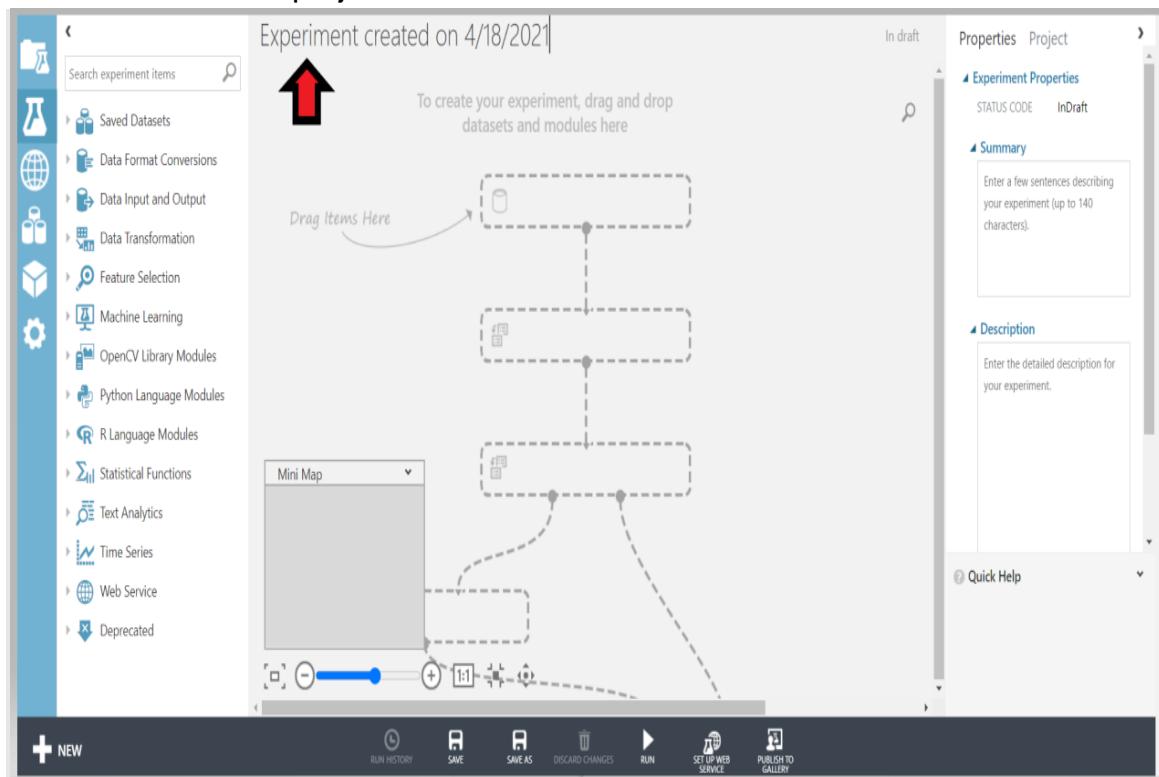
- First select **Experiment** and then **New** at the bottom of the page.



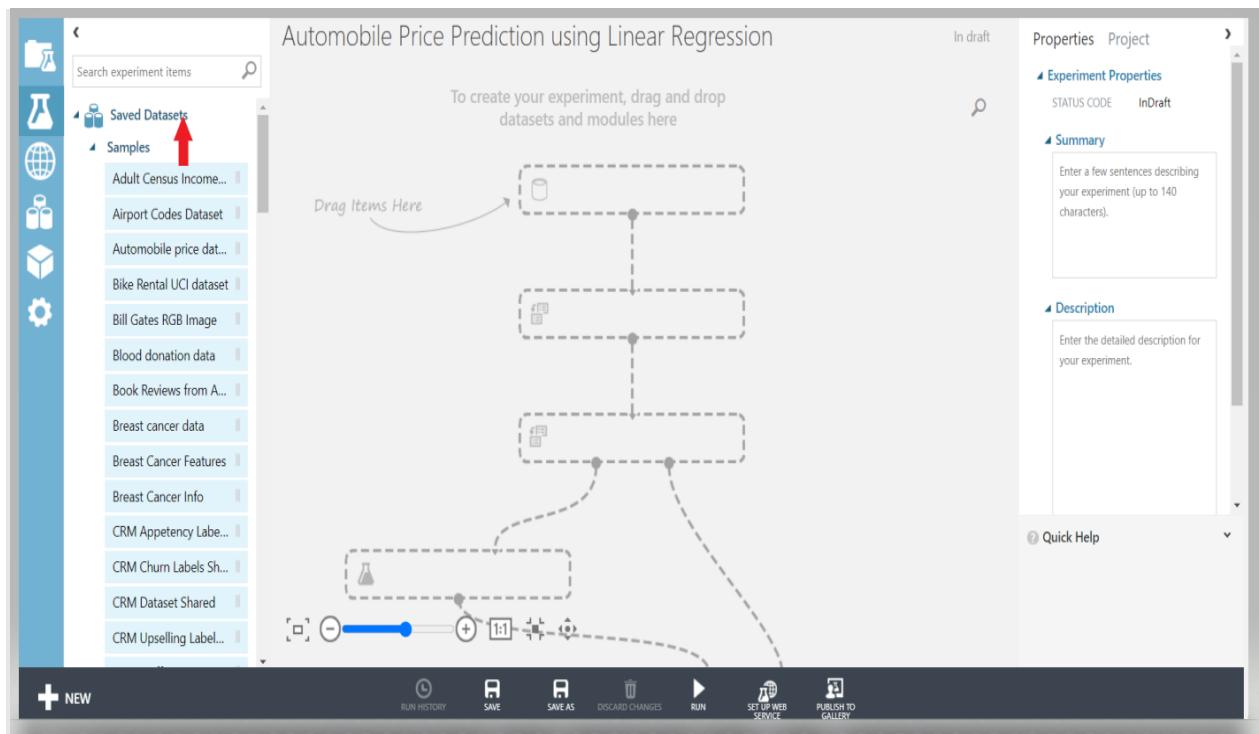
- Select Blank Experiment.



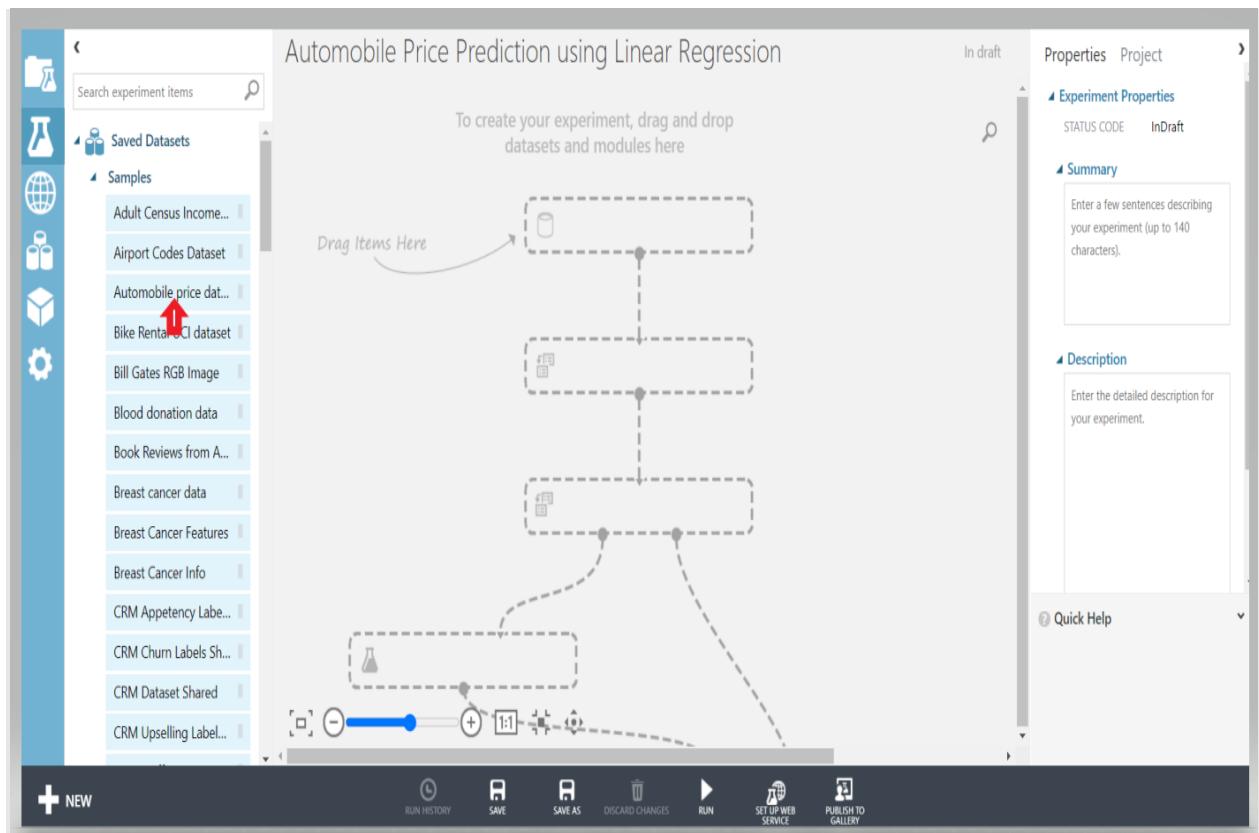
- Give the title for the project.



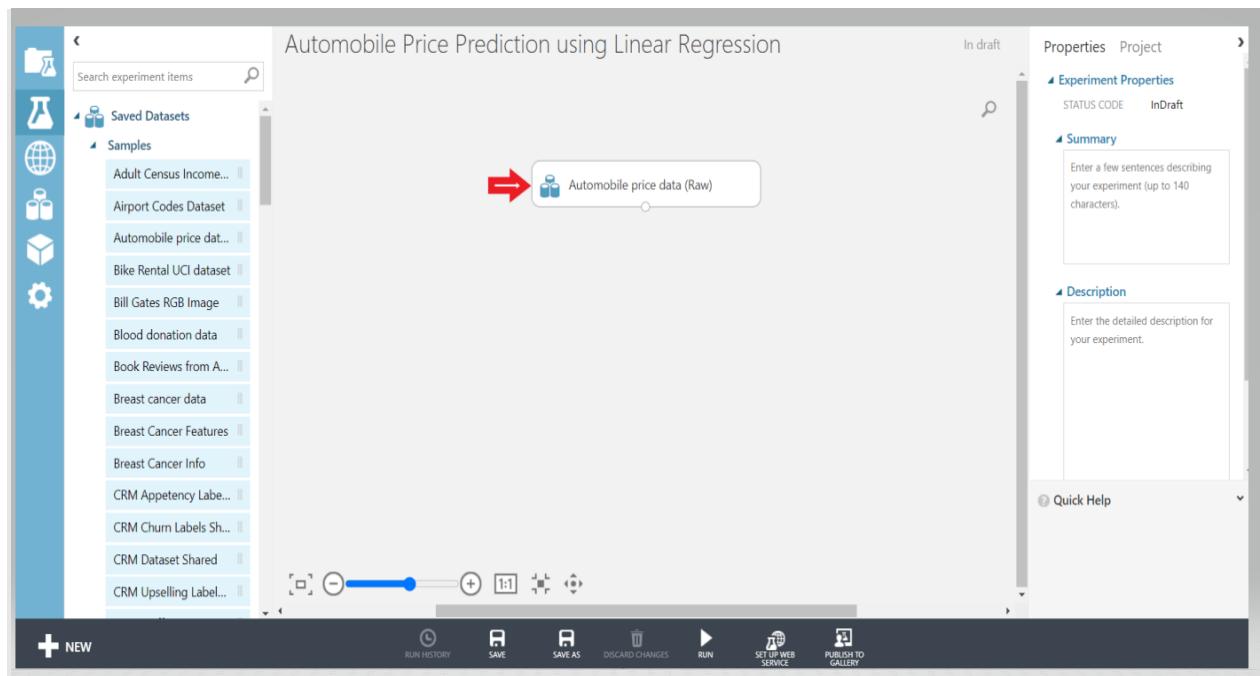
- Select **Sample** option from the **Saved Dataset**.



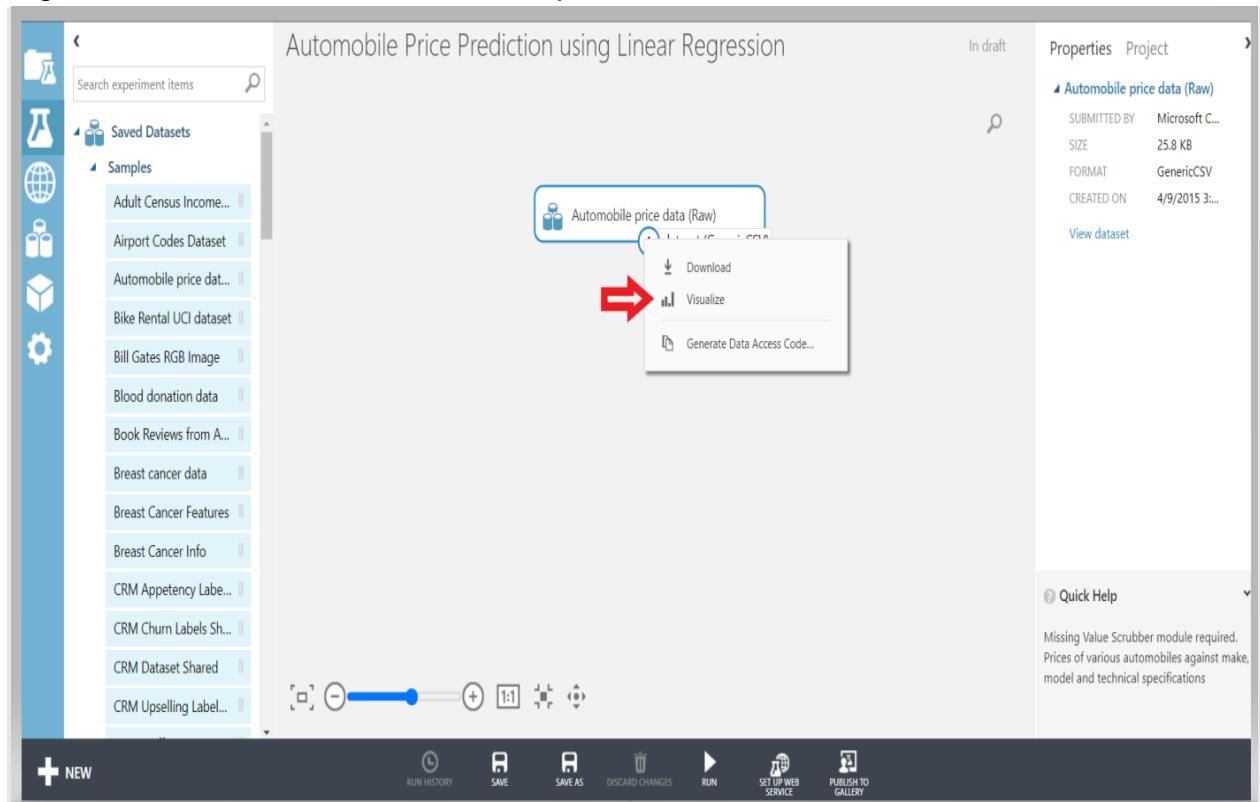
- Select Automobile Price Data Dataset



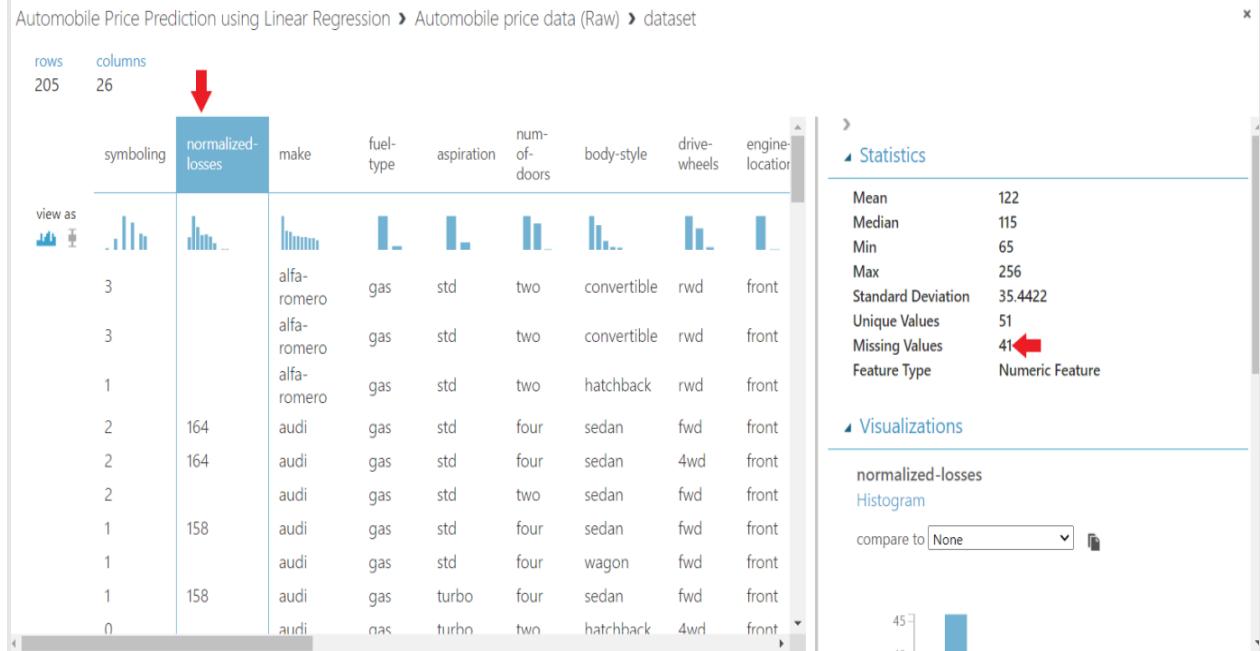
- Drag selected dataset on Panel.



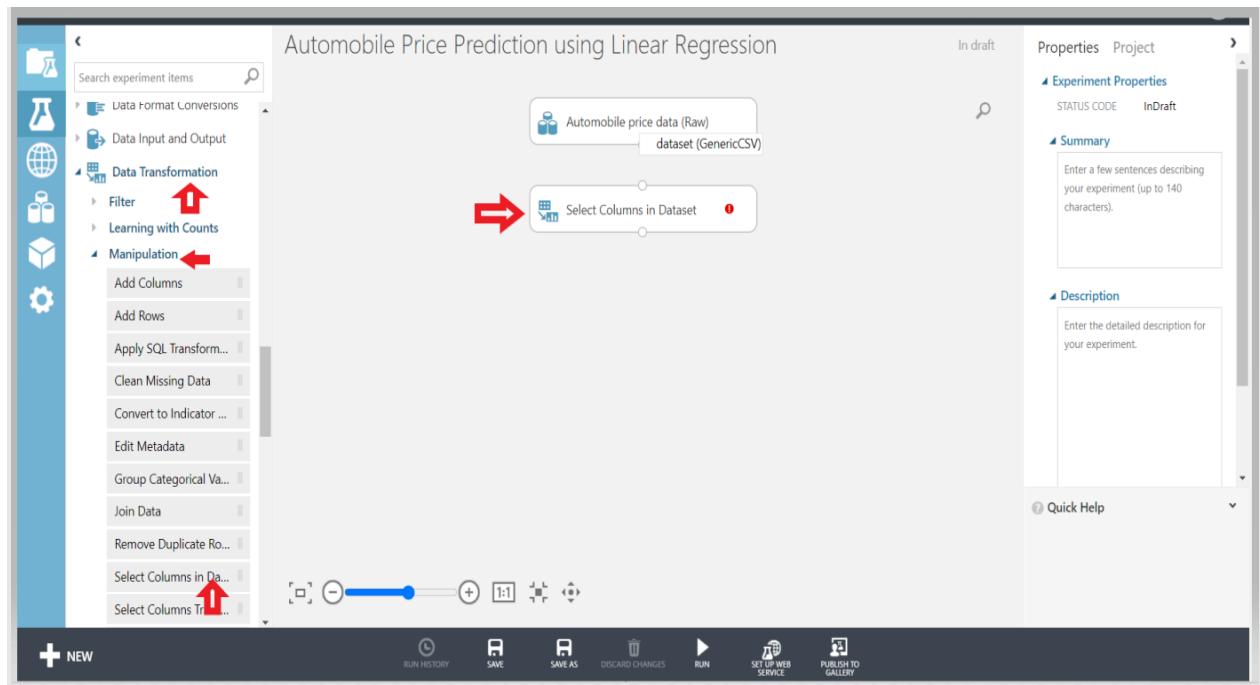
- Right click on 1 and choose visualize option.



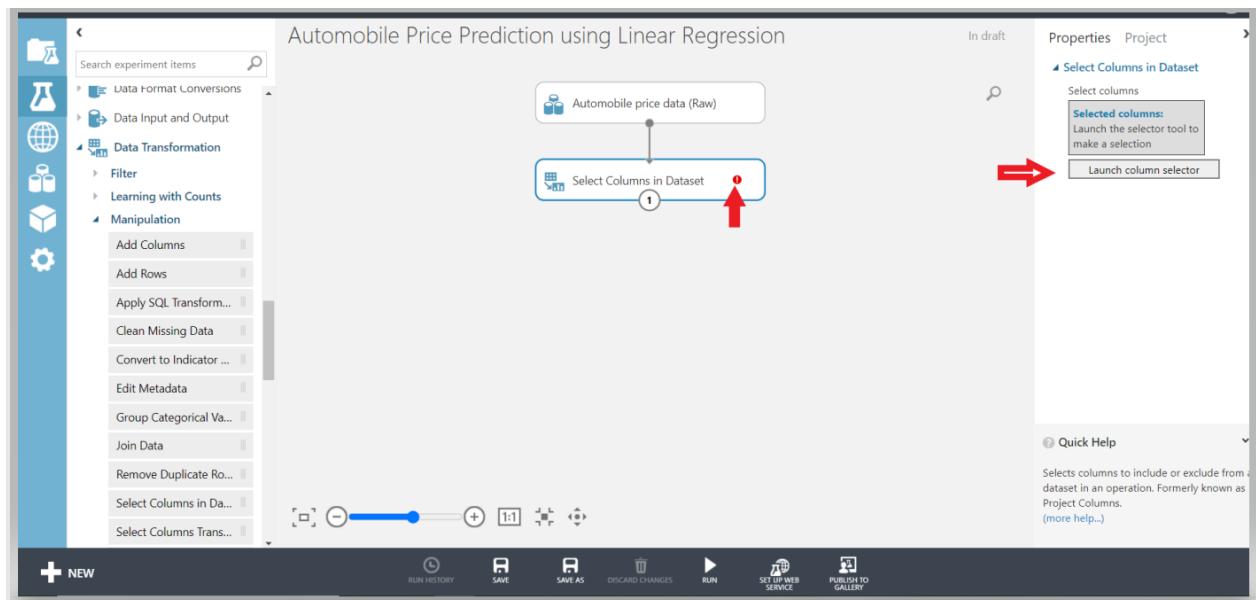
- Visualize the Dataset and then close it.  
For example: Normalized losses has the 41 missing value which is the maximum..



- Select the Data Transformation → Manipulation → Select Columns in Dataset and drag it into Panel.

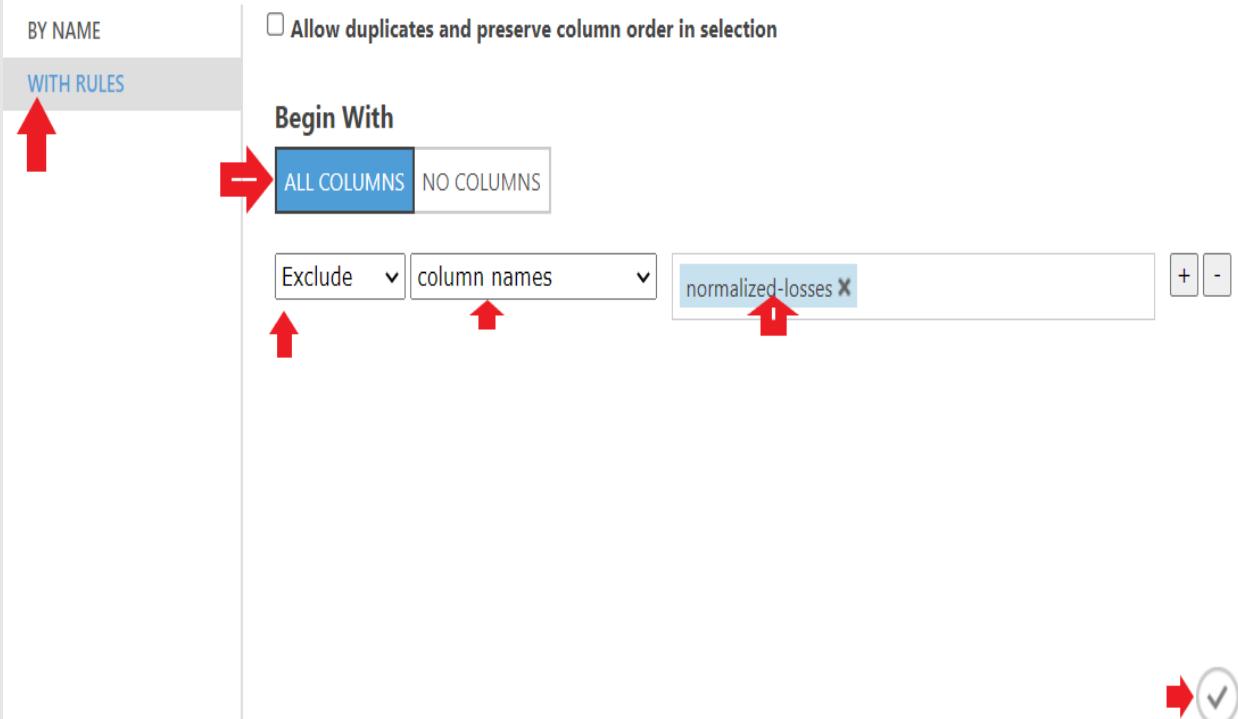


- Make connection between dragged item, press on red sign and then Launch column selector to choose the relevant column.

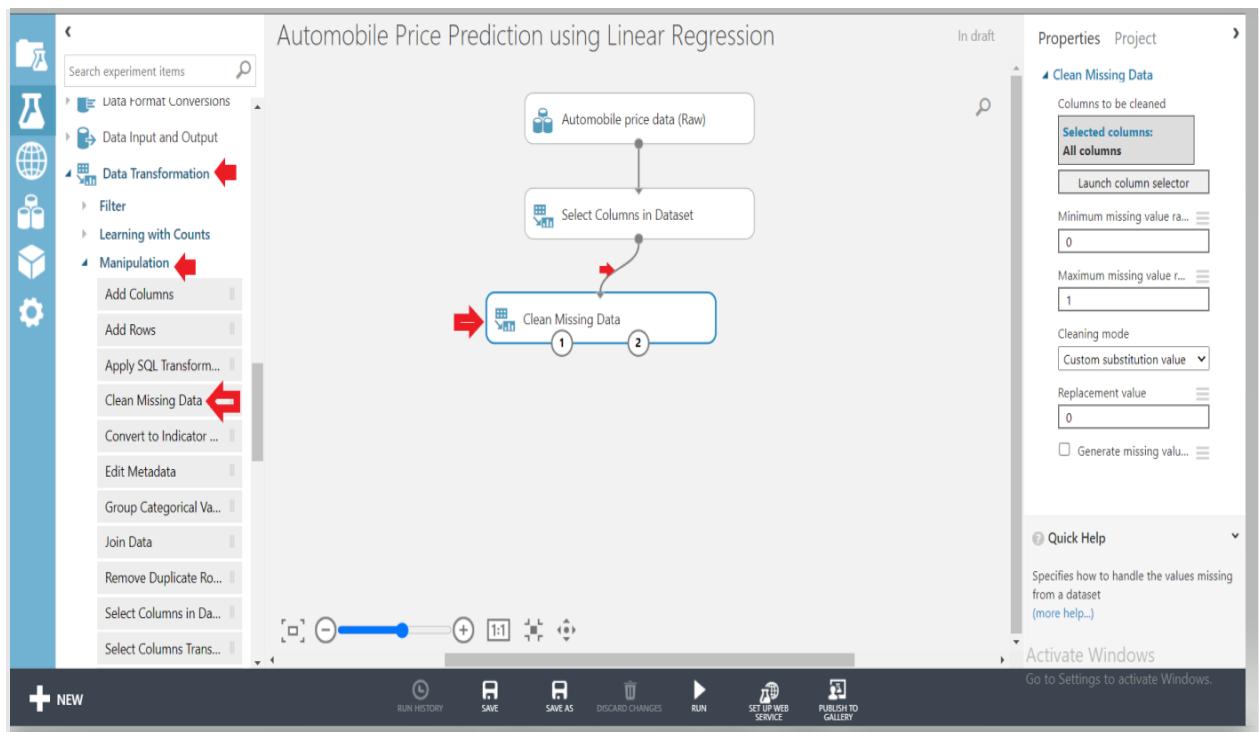


- Remove the column which have the missing value by selecting With Rules → All Columns → Exclude → column names →normalized –losses and click on tick.

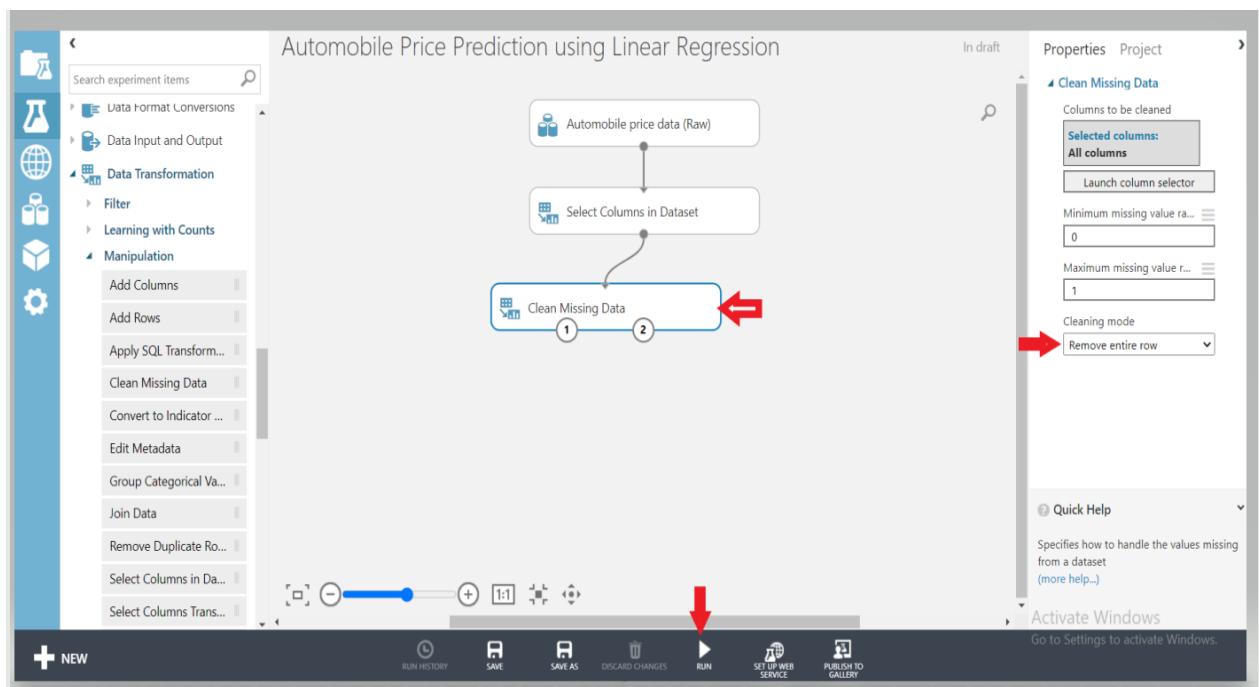
## Select columns



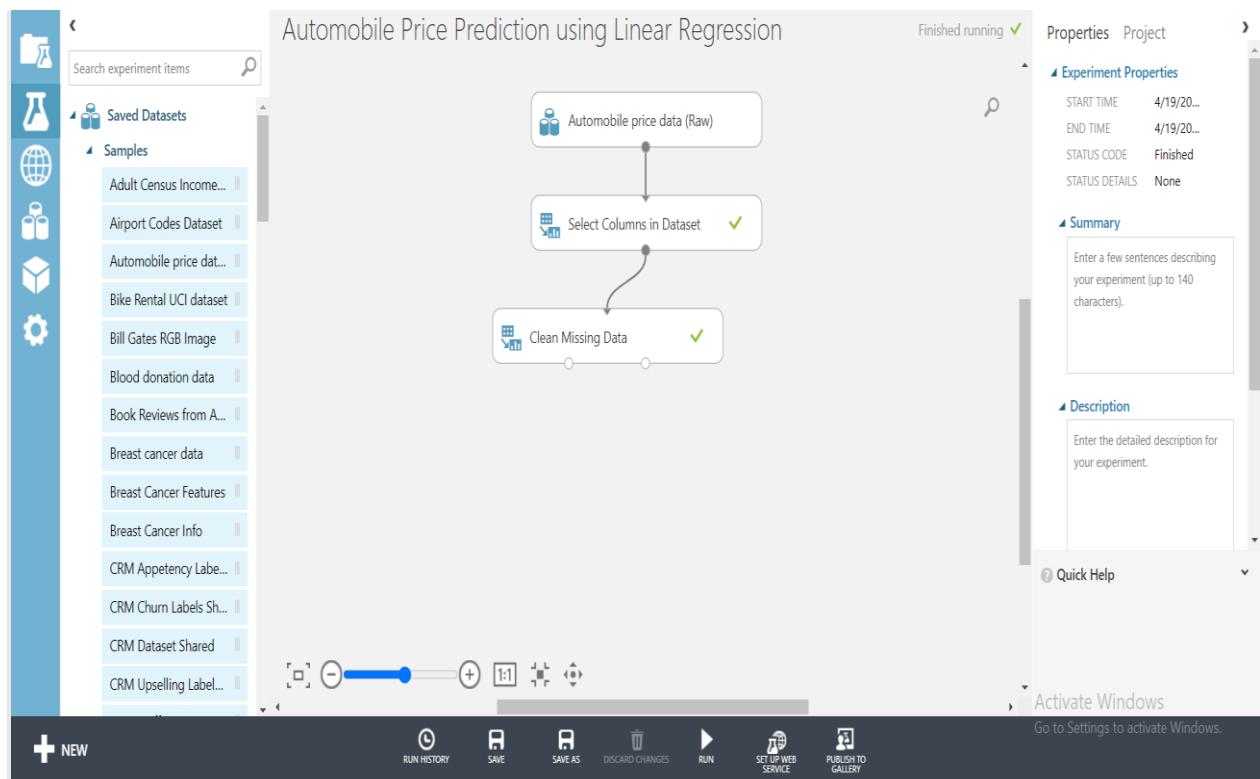
- Select the Data Transformation → Manipulation → Clean missing data, drag it into Panel and make connection.



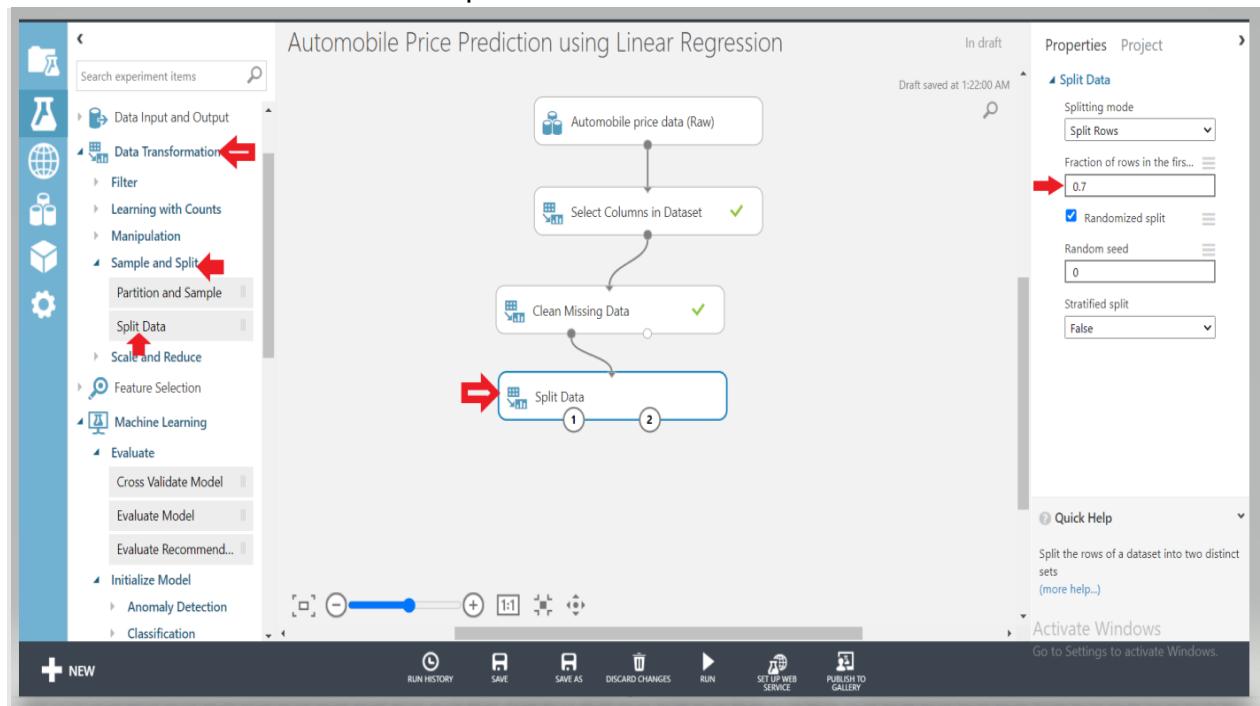
- Click on Clean missing data, set Remove entire row in cleaning mode and press Run.



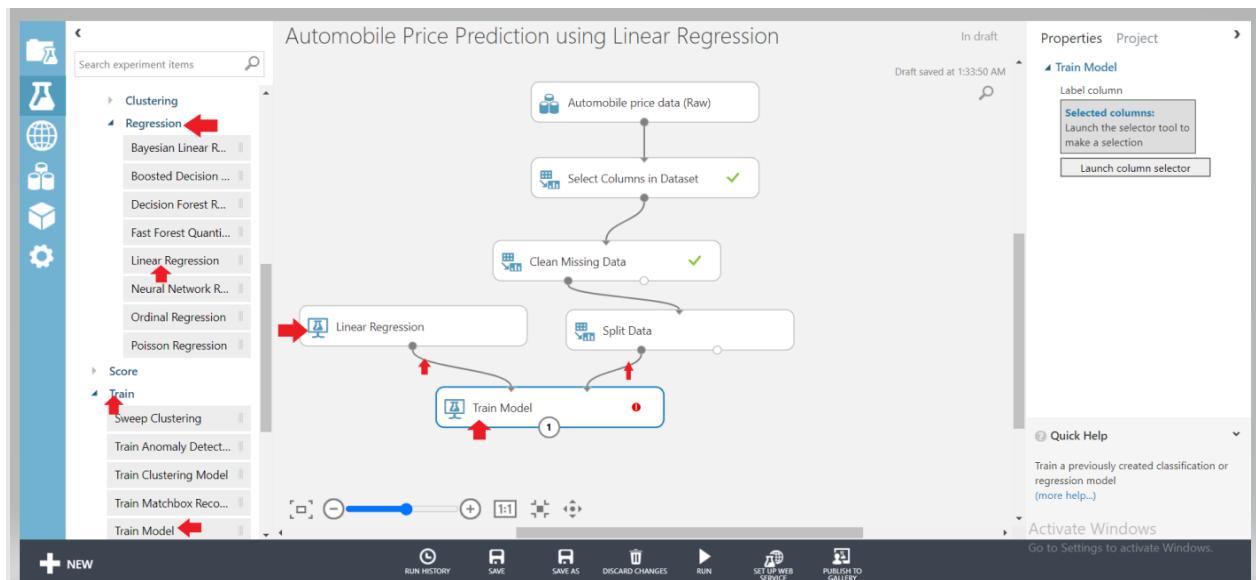
- If there is green tick, it means no error.



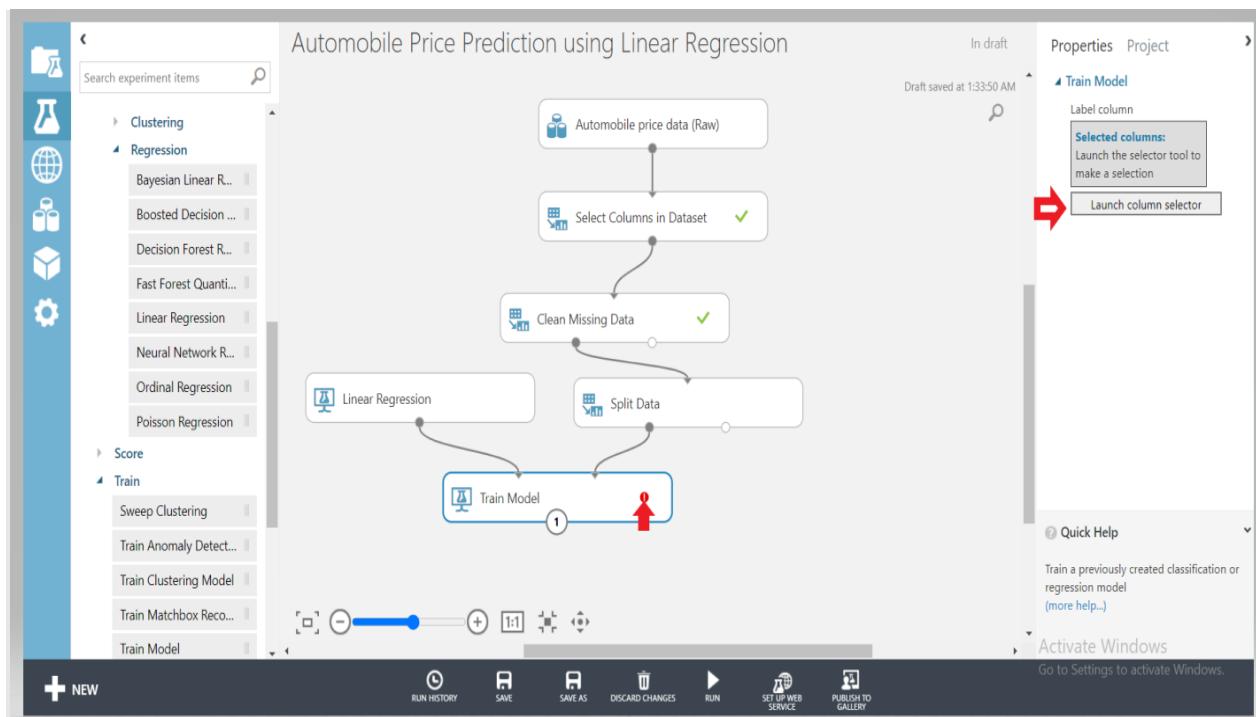
- Select Data Transformation → Sample and Split → Split data and drag it into Panel. Select the value 0.75 to split the data.



- Select Machine Learning → Train → Train Model and drag it into the Panel and connect it.
- Select the machine learning Algorithm from Machine Learning → Initialize Model → Regression → Linear Regression, drag it into panel and make connection.

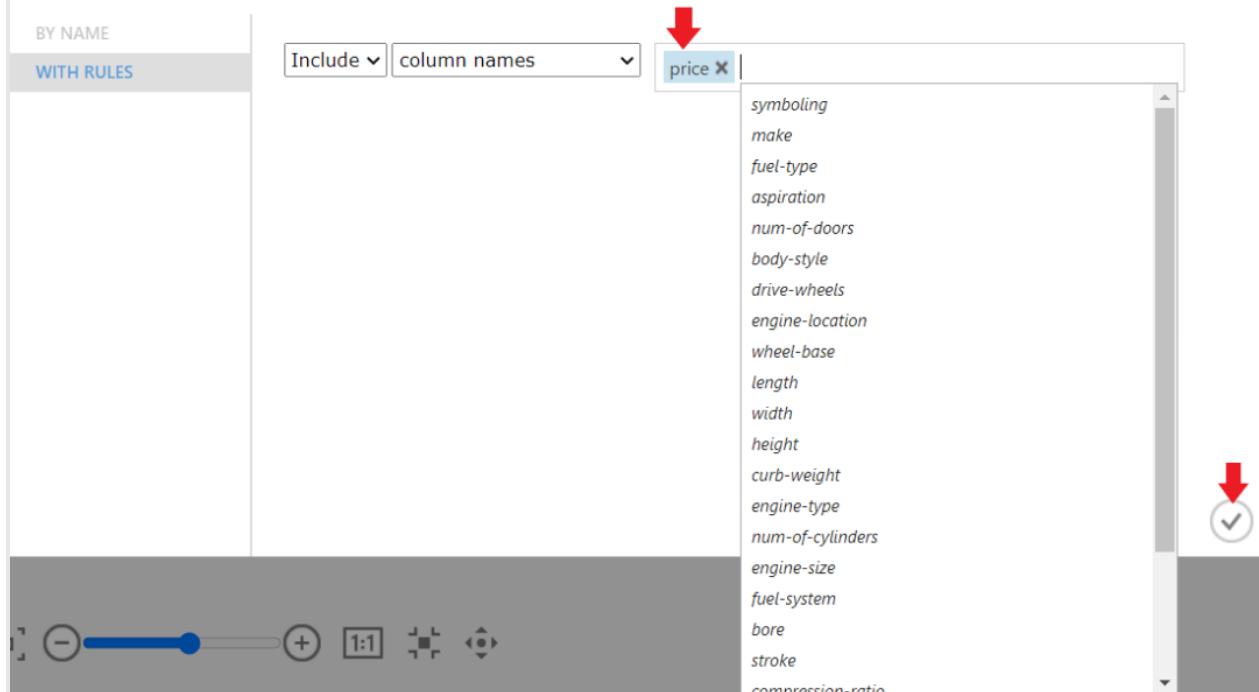


- Click on Train Model and Launch Column Selector



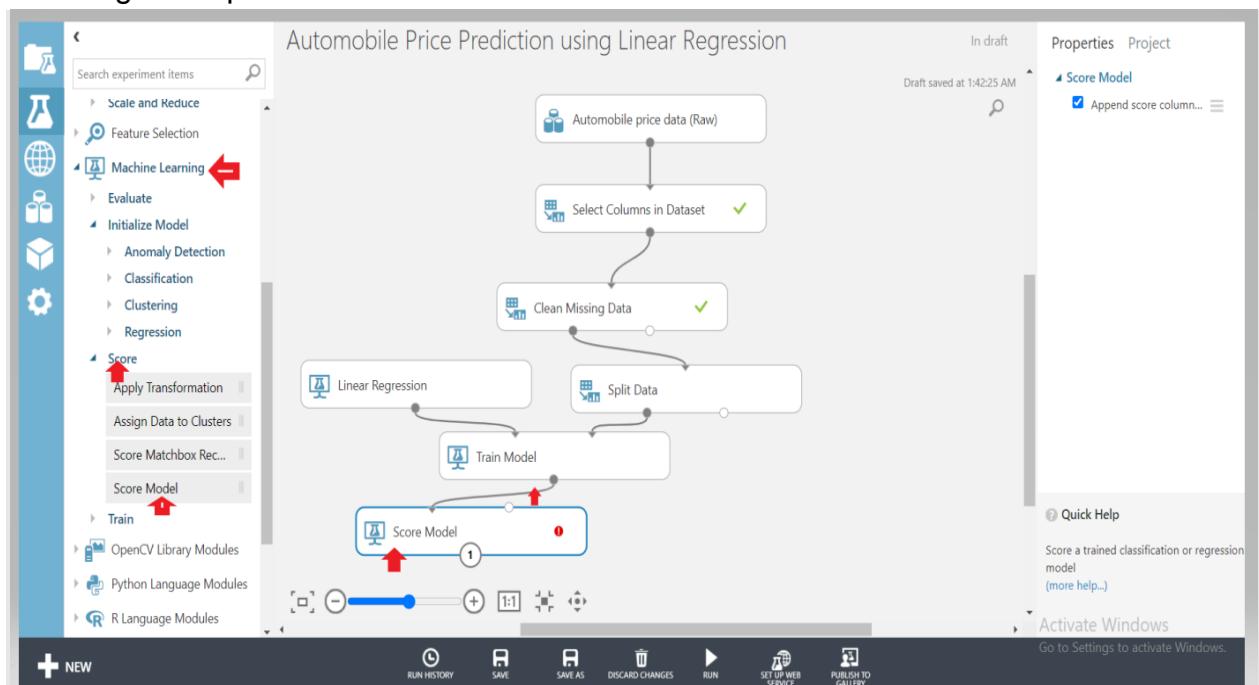
- Now Select price as output for the prediction and press the tick mark.

Select a single column

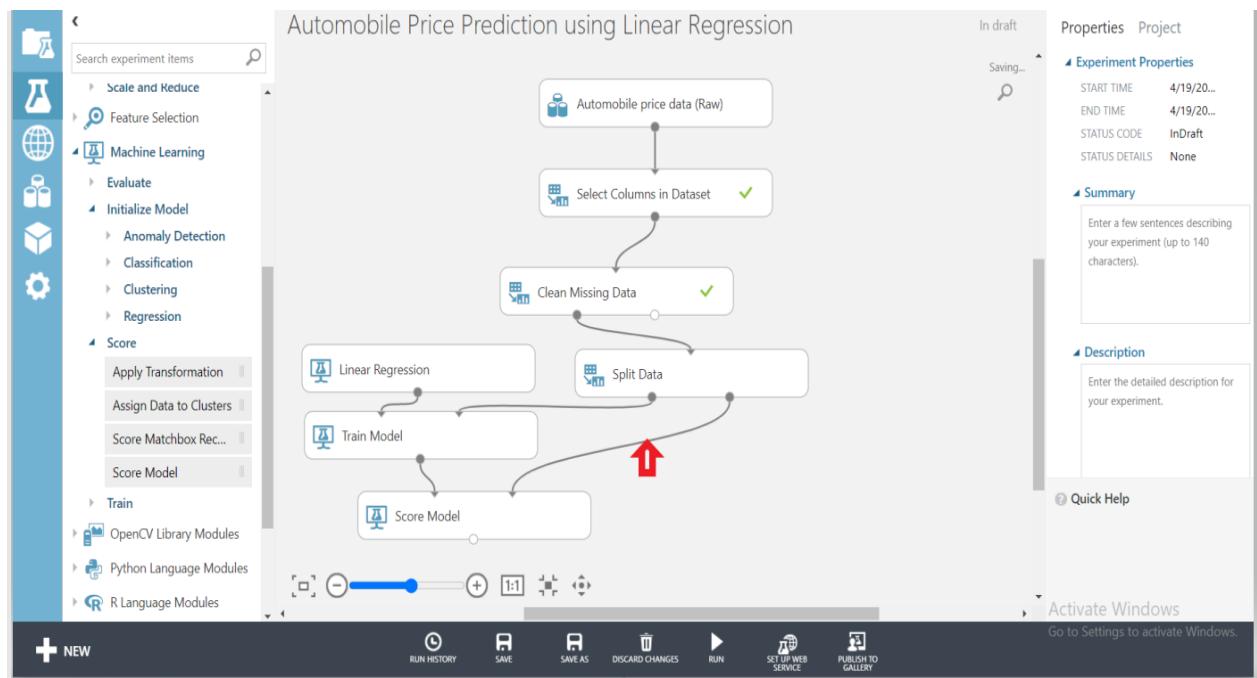


The screenshot shows the 'Select a single column' step in the Microsoft Power BI Data Flow interface. On the left, there's a sidebar with 'BY NAME' and 'WITH RULES' sections. In the center, there's a search bar with 'Include' and 'column names' dropdowns, and a list of columns on the right. A red arrow points to the word 'price' in the list. At the bottom right of the list, there's a checkmark icon with a red arrow pointing to it. The bottom of the screen has a toolbar with various icons.

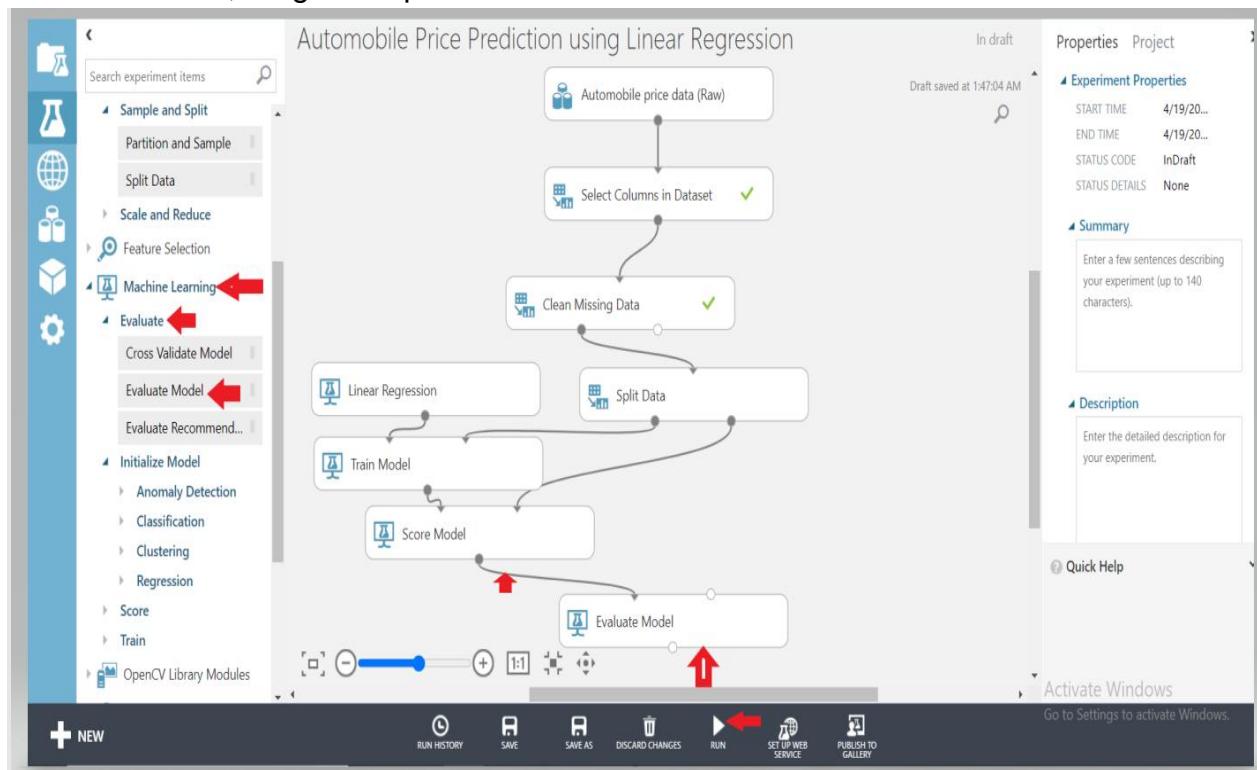
- For Score calculation, Choose Machine Learning → Score → Score Model and drag it into panel and connect it.



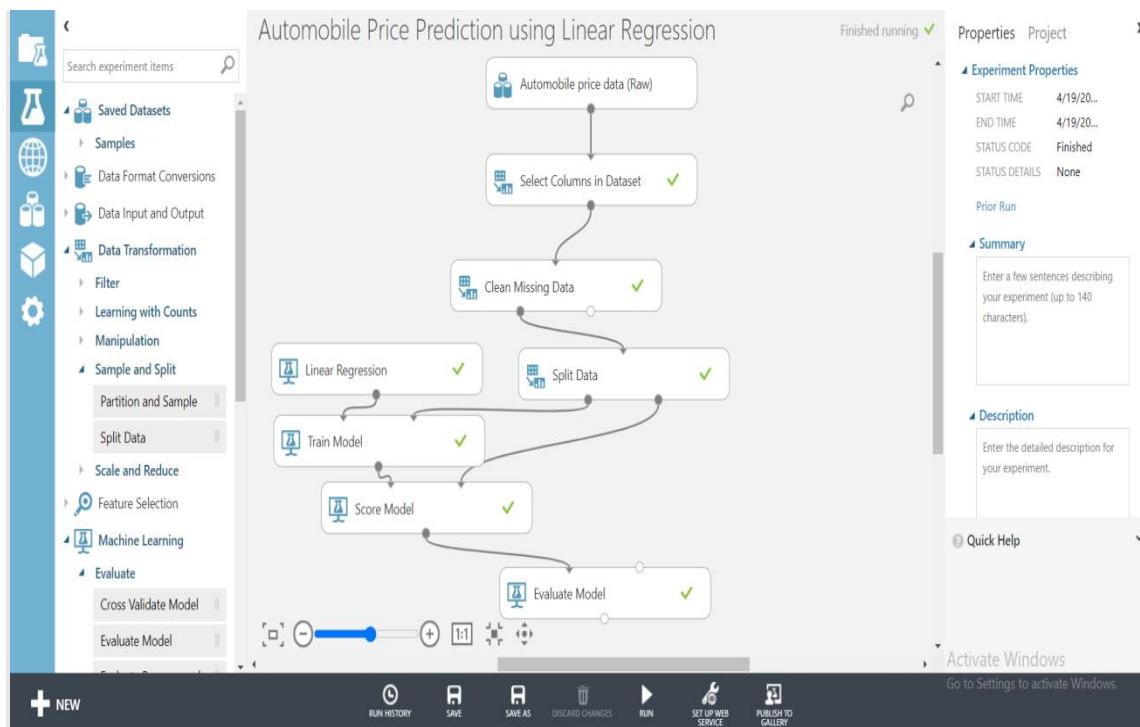
- Connect Split Data with Score Model for Testing.



- For Evaluation of the model choose, Machine Learning → Evaluate → Evaluate Model, drag it into panel and connect. Run the model.



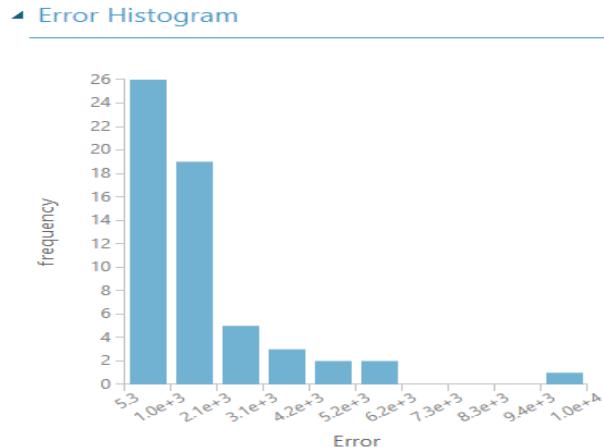
- Green tick shows that model has run successfully. Entire model would be display as below:



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

Automobile Price Prediction using Linear Regression > Evaluate Model > Evaluation results

Metrics	
Mean Absolute Error	1605.514464
Root Mean Squared Error	2385.271889
Relative Absolute Error	0.2666248
Relative Squared Error	0.083112
Coefficient of Determination	0.916888



We just explored the Microsoft Azure machine learning studio which is GUI based integrated development environment for constructing and operationalizing machine learning workflow and we also constructed linear

regression model. Now let us get started with logistic regression algorithm which is mainly used for classification.

## 4.10 Logistic Regression

- 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:

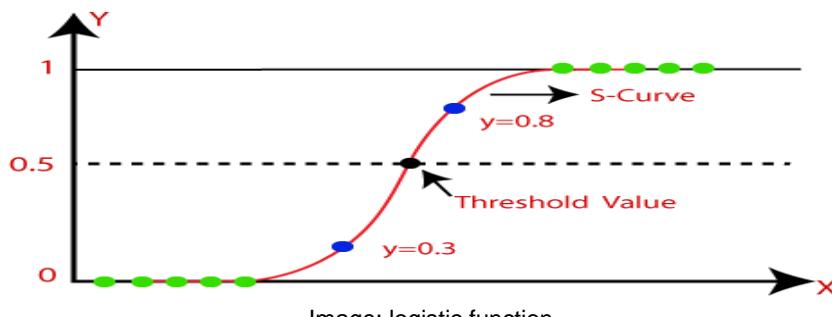


Image: logistic function

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  $-\infty$  to  $+\infty$ , 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.

## Type 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".

## Estimating probabilities

Probability is one of the foundations of machine learning (along with linear algebra and optimization). Probability is a measure of uncertainty. Probability applies to machine learning because in the real world, we need to make decisions with incomplete information. Hence, we need a mechanism to quantify uncertainty – which Probability provides us. Using probability, we can model elements of uncertainty such as risk in financial transactions and many other business processes. In contrast, in traditional programming, we work with deterministic problems i.e., the solution is not affected by uncertainty.

## Probability of an event

Probability quantifies the likelihood or belief that an event will occur. Probability theory has three important concepts: Event - an outcome to which a probability is assigned; The Sample Space which represents the set of possible outcomes for the events and the Probability Function which maps a probability to an event. The probability function indicates the likelihood that the event being a part of the sample space is drawn. The probability distribution represents the shape or distribution of all events in the sample space. The probability of an event can be calculated directly by counting all the occurrences of the event and dividing them by the total possible outcomes of the event. Probability is a fractional value and has a value in the range between 0 and 1, where 0 indicates no probability and 1 represents full probability.

## Applications

It explores how probability can apply to machine learning

### 1. Sampling - Dealing with non-deterministic processes

Probability forms the basis of sampling. In machine learning, uncertainty can arise in many ways – for example - noise in data. Probability provides a set of tools to model uncertainty. Noise could arise due to variability in the observations, as a measurement error or from other sources. Noise effects both inputs and outputs.

Typically, we are given a dataset i.e., we do not have control on the creation and sampling process of the dataset. To cater for this lack of control over sampling, we split the data into train and test sets or we use resampling techniques. Hence,

probability (through sampling) is involved when we have incomplete coverage of the problem domain.

## 2. Pattern recognition

Pattern recognition is a key part of machine learning. We can approach machine learning as a pattern recognition problem from a Bayesian standpoint. In Pattern Recognition – one takes a Bayesian view and presents approximate inference algorithms for situations where exact answers are not feasible. For the same reasons listed above, Probability theory is a key part of pattern recognition because it helps to cater for noise / uncertainty and for the finite size of the sample and also to apply Bayesian principles to machine learning.

## 3. Training - use in Maximum likelihood estimation

Many iterative machine learning techniques like Maximum likelihood estimation (MLE) are based on probability theory. MLE is used for training in models like linear regression, logistic regression and artificial neural networks.

## 4. Developing specific algorithms

Probability forms the basis of specific algorithms like Naive Bayes classifier

## 5. Hyperparameter optimization

In machine learning models such as neural networks, hyperparameters are tuned through techniques like grid search. Bayesian optimization can be also used for hyperparameter optimization.

## 6. Model evaluation

In binary classification tasks, we predict a single probability score. Model evaluation techniques require us to summarize the performance of a model based on predicted probabilities. For example – aggregation measures like log loss require the understanding of probability theory.

## Project : Logistic Regression using Sklearn

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.

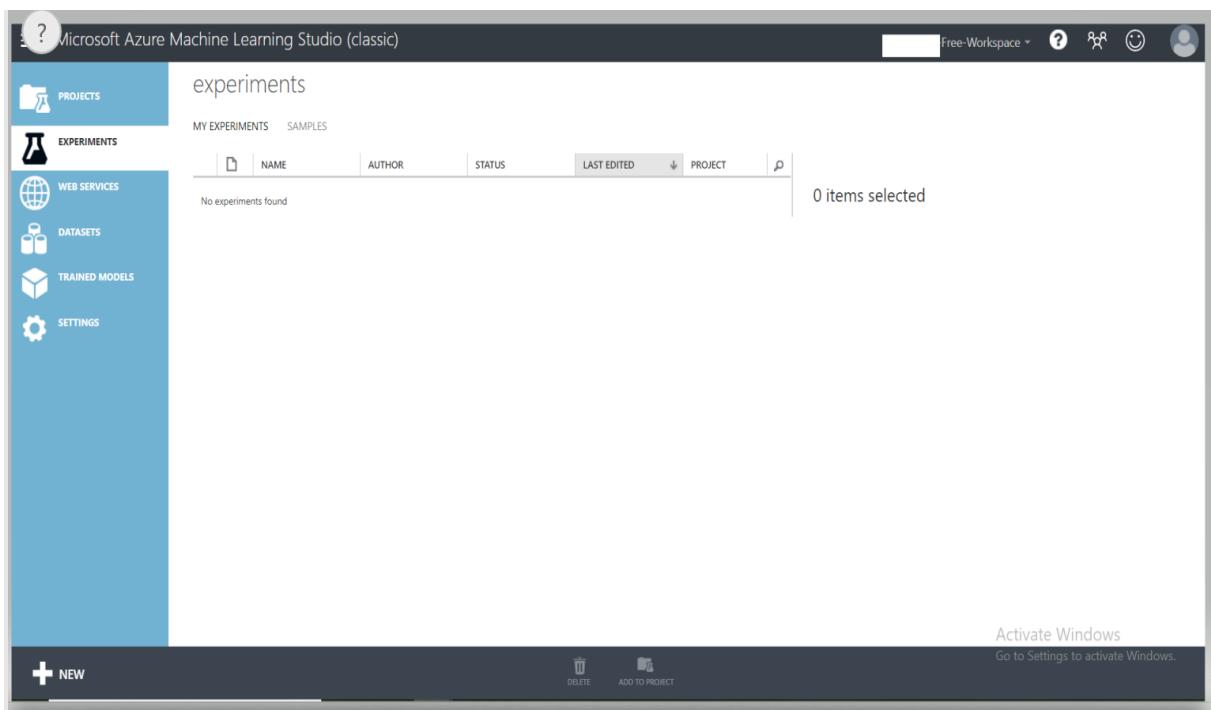
The Link for project is given below:

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/2.Logistic%20Regression.ipynb>

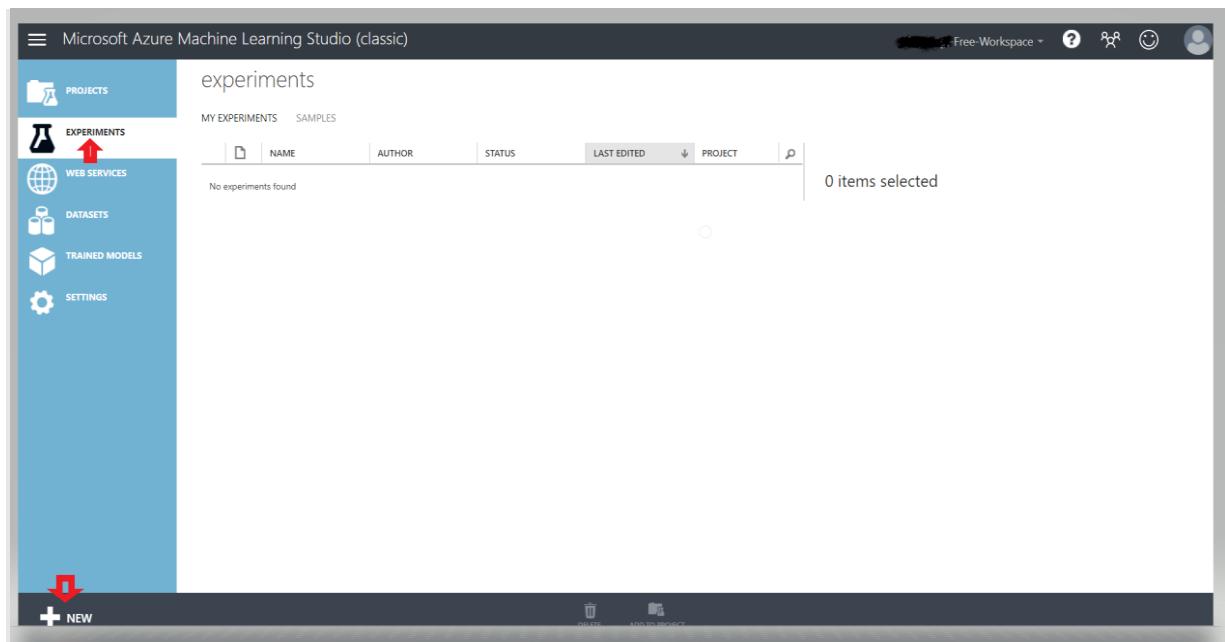
After understanding the fundamentals of logistics regression, let us get started to build model using ML studio.

## 4.11 Logistic Regression model with ML Studio

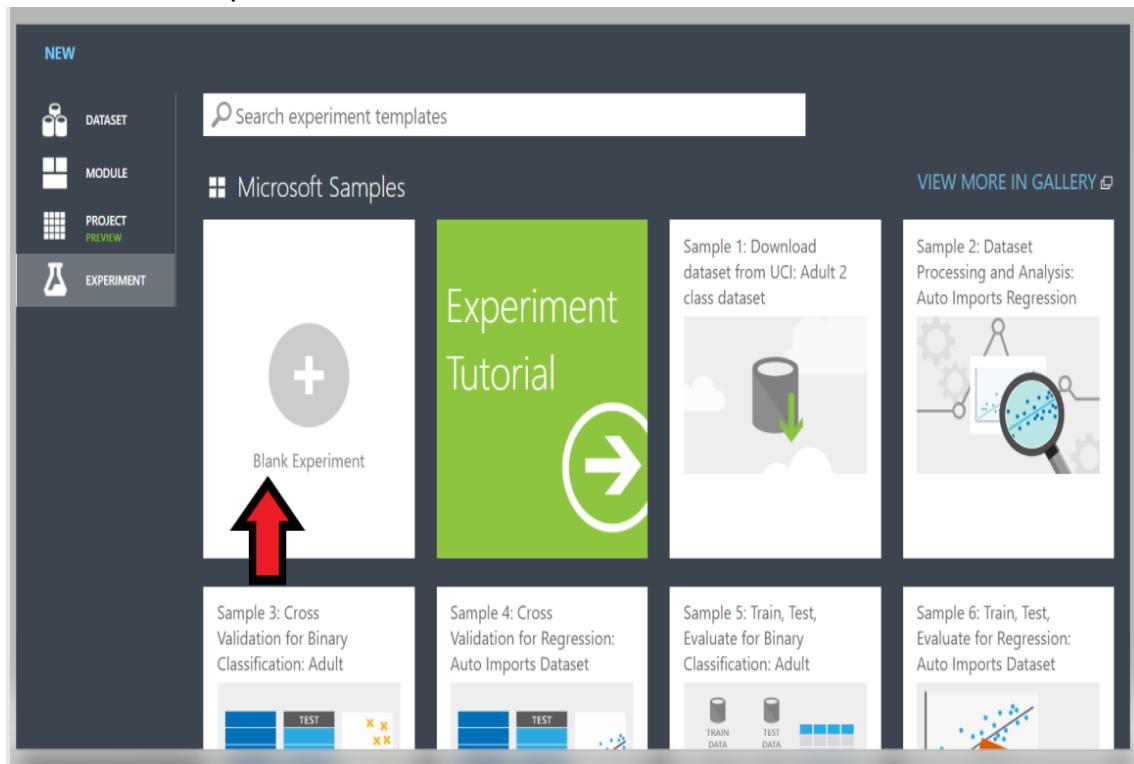
- Sign in into Microsoft Azure Machine Learning Studio (classic) and create workspace.



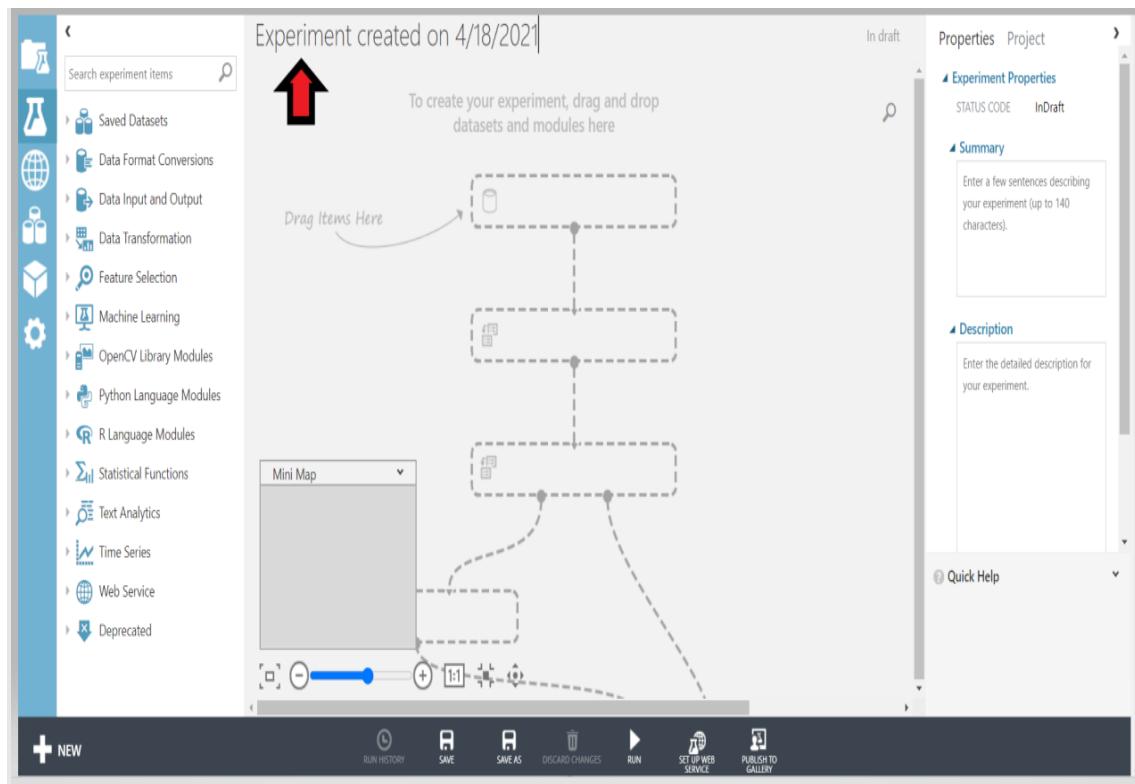
- First select **Experiment** and then **New** at the bottom of the page.



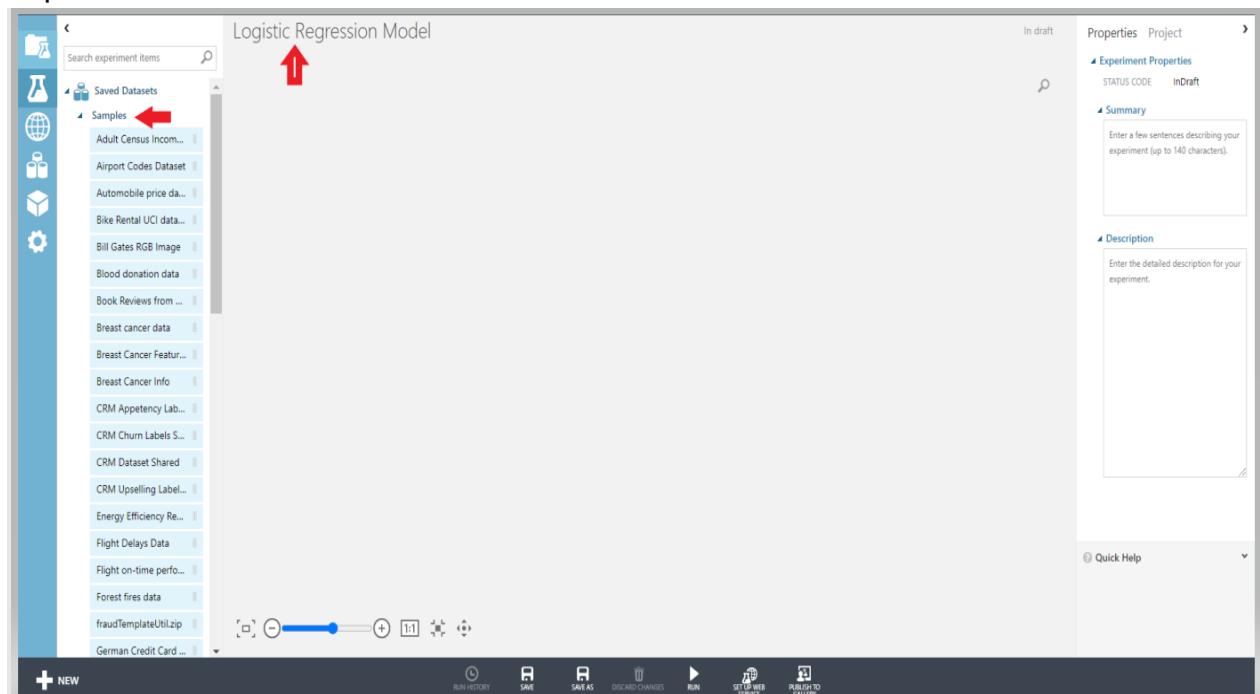
- Select Blank Experiment.



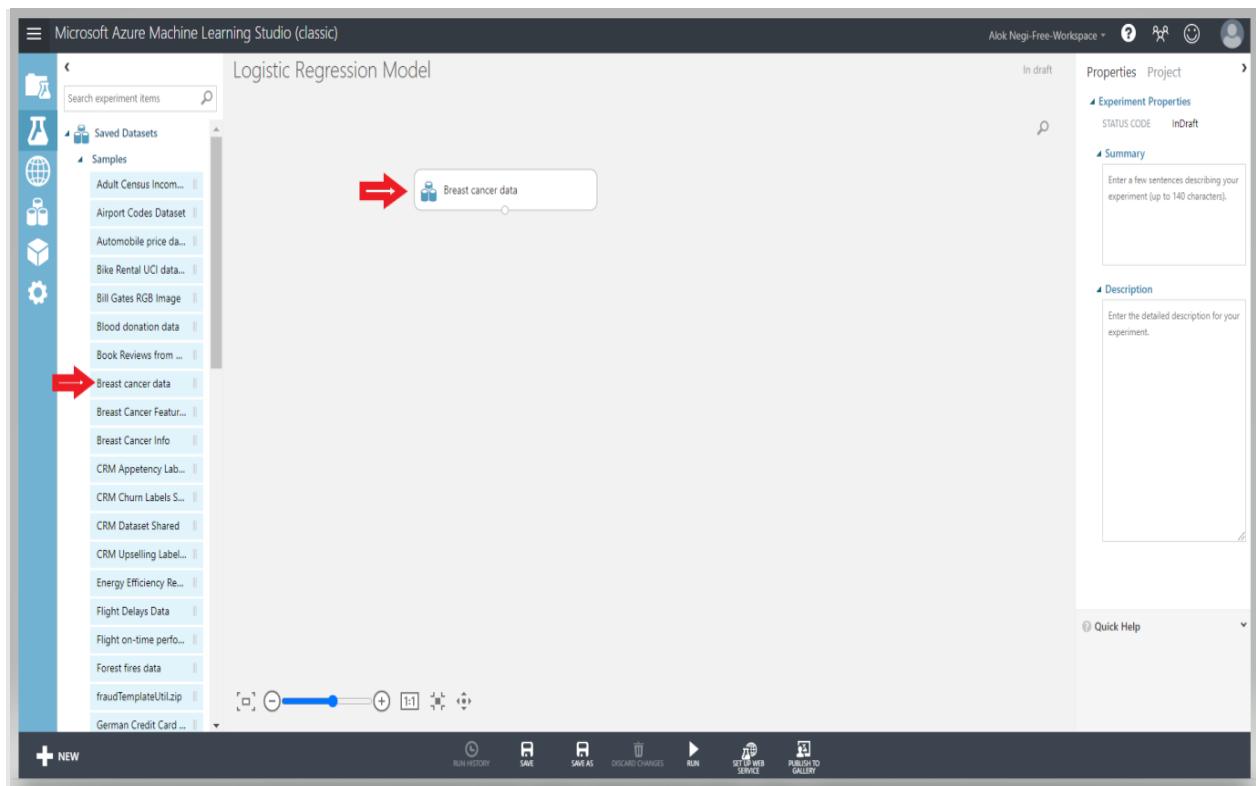
- Give the title for the project.



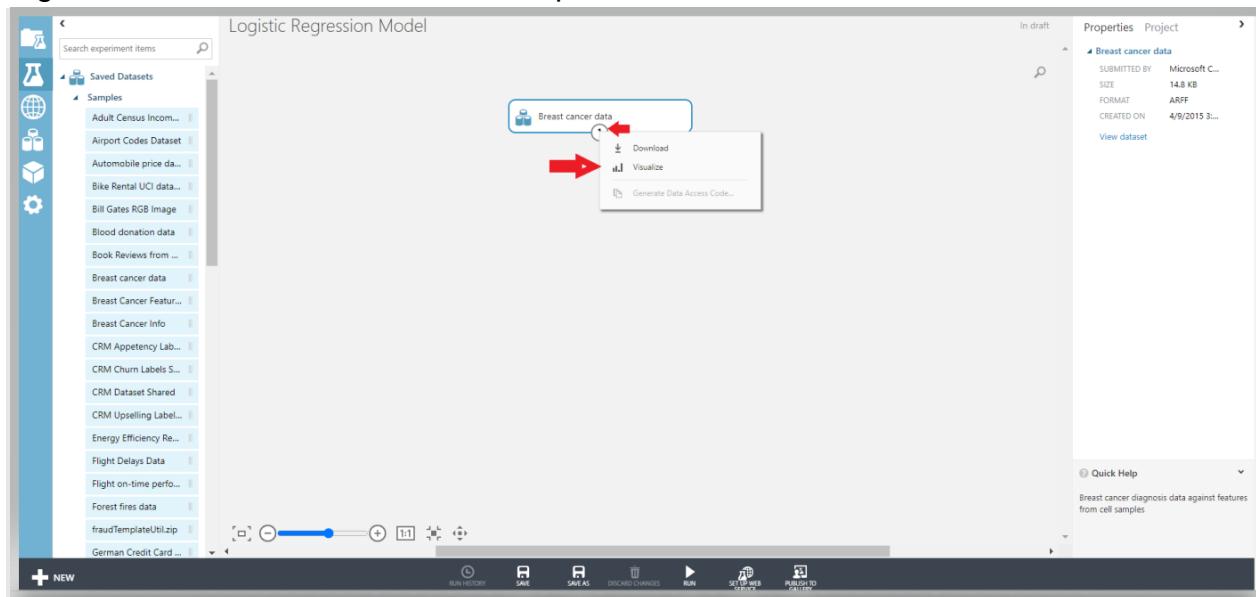
- Select **Sample** option from the **Saved Dataset**. Check the Title for the Experiment.



- Select Breast Cancer Dataset and Drag selected dataset on Panel..

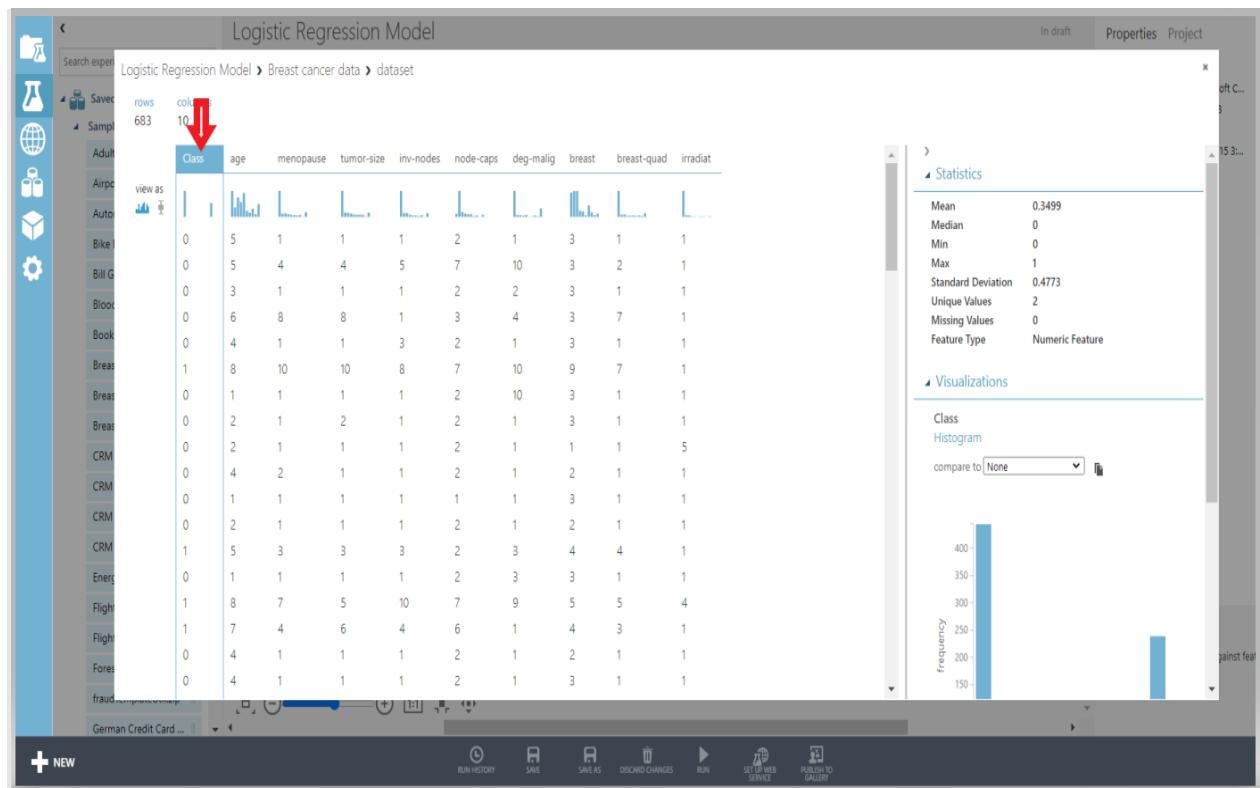


- Right click on 1 and choose visualize option.

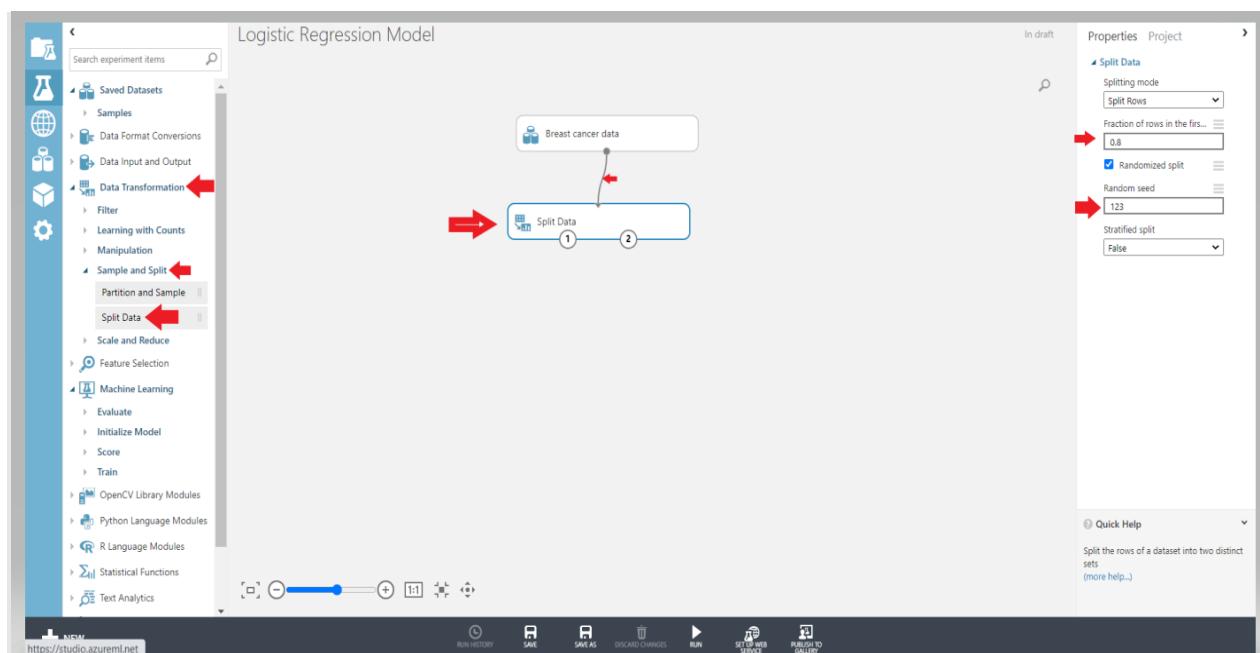


- Visualize the Dataset and then close it.

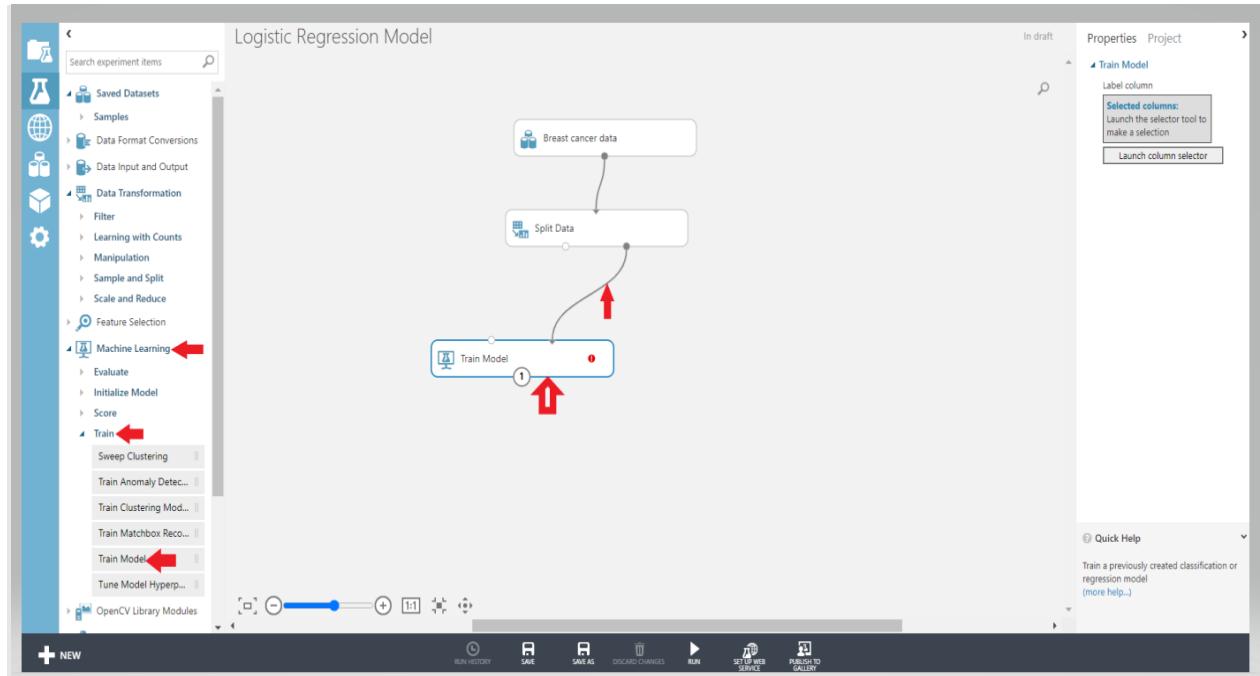
For example: In Class column, 0 represents the no cancer while 1 represents the cancer.



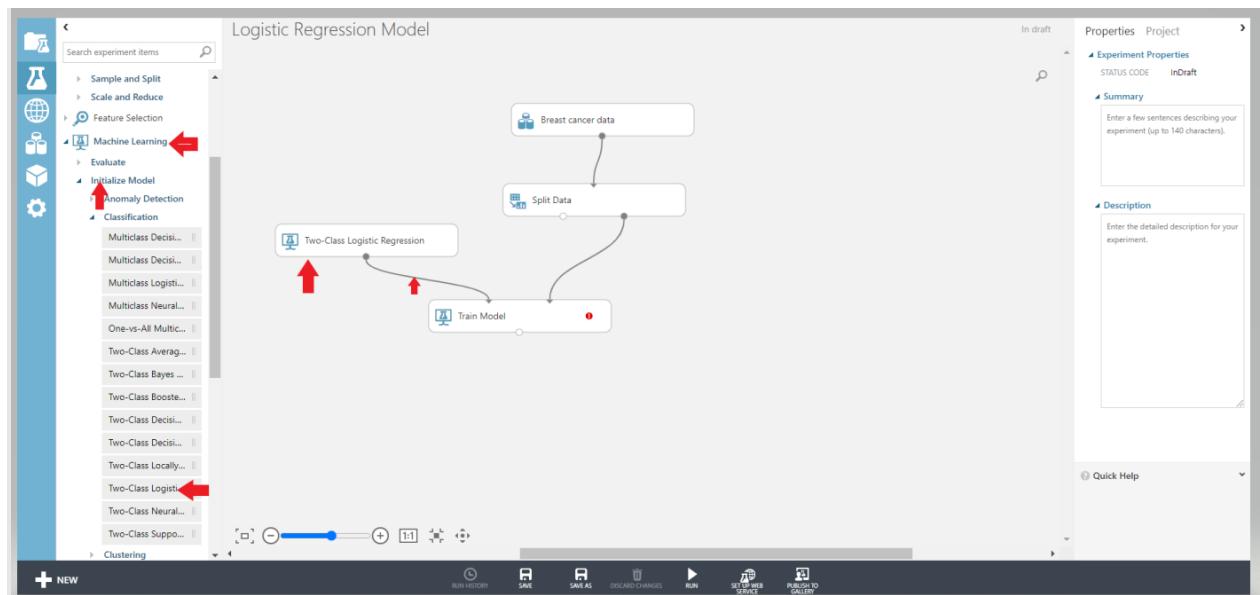
- Select Data Transformation → Sample and Split → Split data, drag it into Panel and connect. Select the value 0.8 to split the data between train and test. Set Random seed = 123.



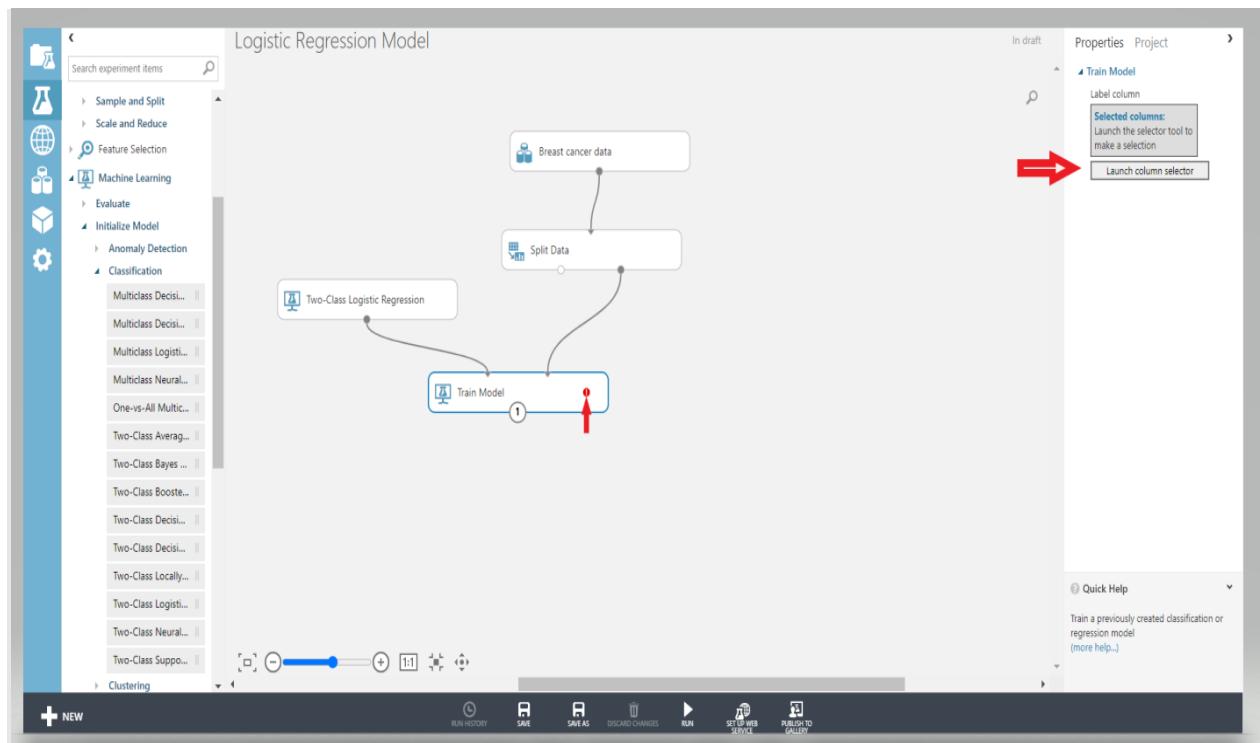
- Select Machine Learning → Train → Train Model and drag it into the Panel and connect it.



- Select the machine learning Algorithm from Machine Learning → Initialize Model →Classification → Two Class Logistic Regression, drag it into panel and make connection.

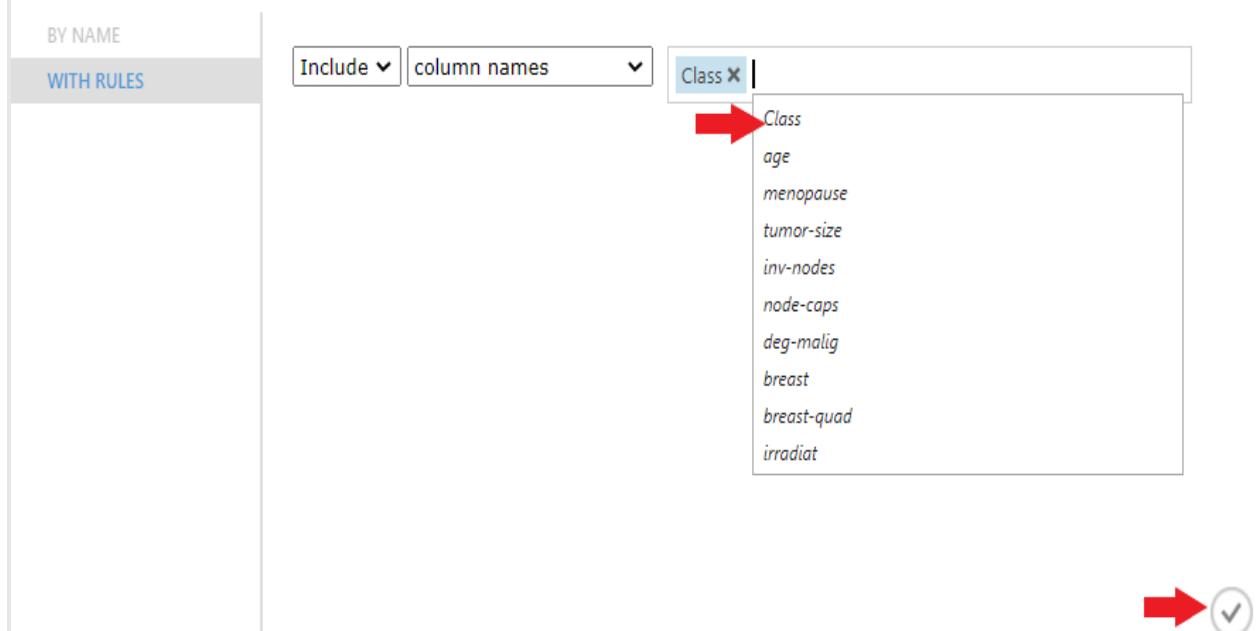


- Click on Train Model and Launch Column Selector



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

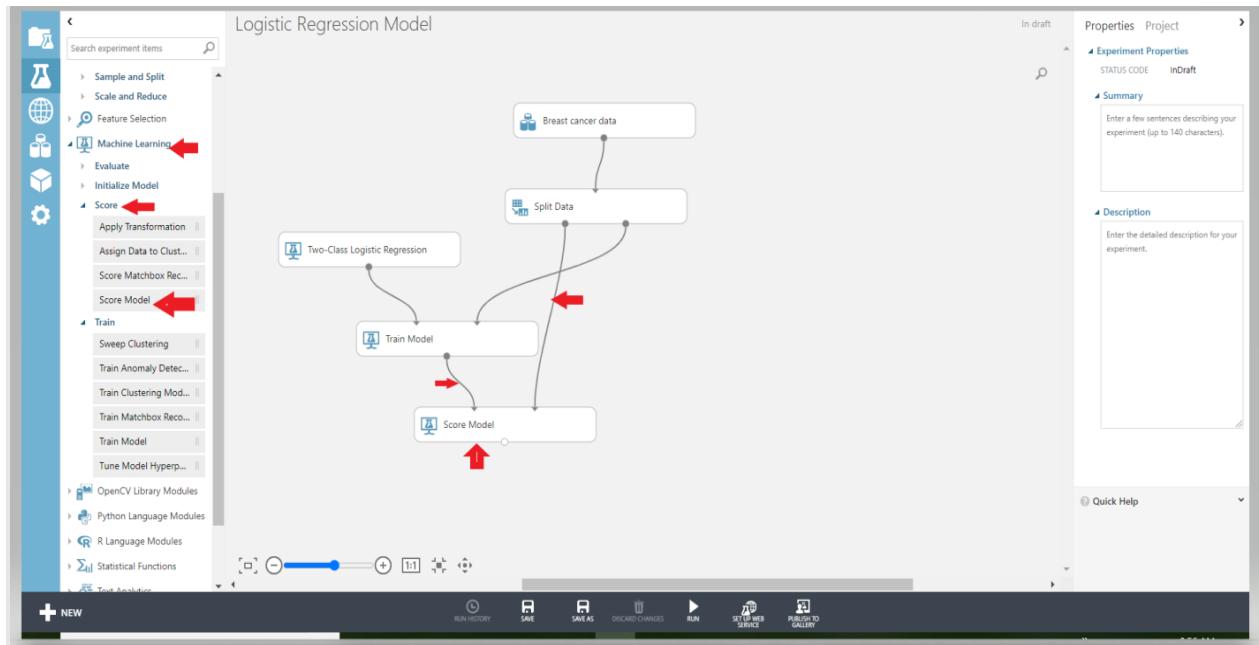
Select a single column



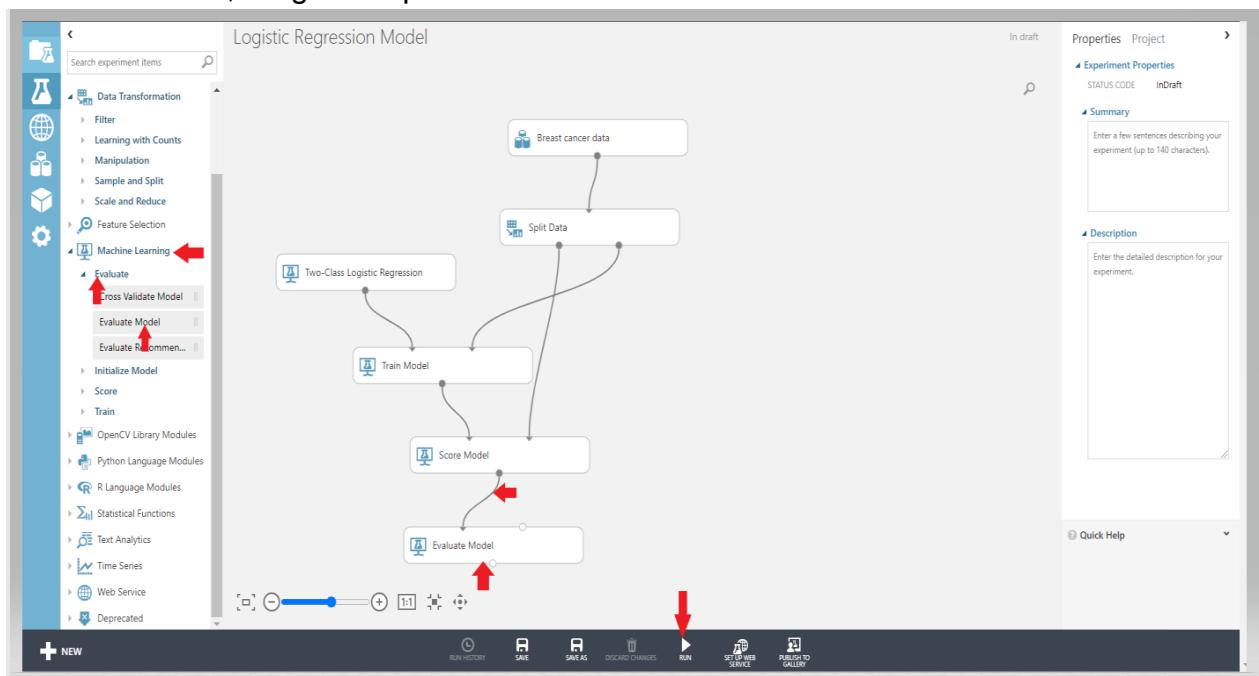
The screenshot shows a 'Select a single column' dialog box. On the left, there are two tabs: 'BY NAME' and 'WITH RULES', with 'WITH RULES' selected. In the center, there's a dropdown menu with 'Include' and 'column names' options. To the right, a list of columns is shown, starting with 'Class'. A red arrow points to the 'Class' entry. At the bottom right of the dialog, there's a checkmark icon with a red arrow pointing to it, indicating the selection is confirmed.

Column
Class
age
menopause
tumor-size
inv-nodes
node-caps
deg-malig
breast
breast-quad
irradiat

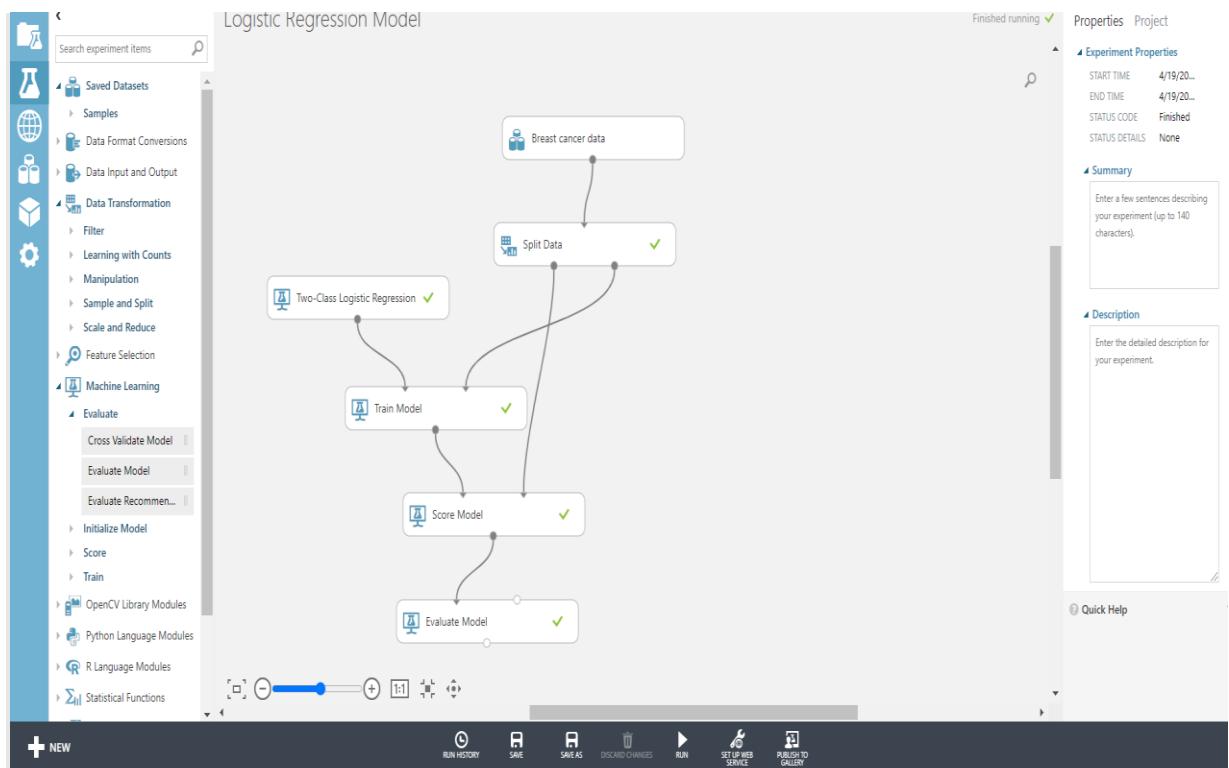
- For Score calculation, Choose Machine Learning → Score → Score Model and drag it into panel and connect it. Also Connect Split Data with Score Model for Testing.



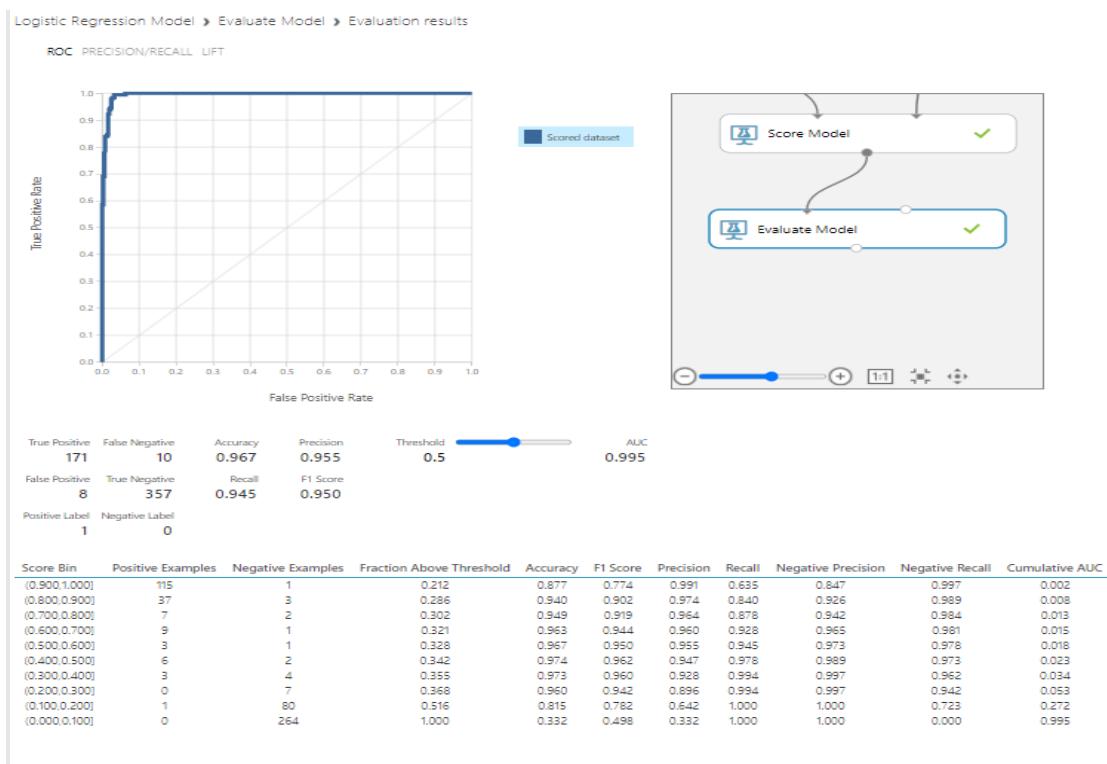
- For Evaluation of the model choose, Machine Learning → Evaluate → Evaluate Model, drag it into panel and connect. Run the model.



- Green tick shows that model has run successfully. Entire model would be display as below:



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



Now, after creating a model on azure platform let us learn about how we can improve the performance of model using the process called fine tuning.

## 4.12 GridSearchCV for Hyperparameter Tuning

Grid search is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. The performance of a model significantly depends on the value of hyperparameters. Note that there is no way to know in advance the best values for hyperparameters so ideally, we need to try all possible values to know the optimal values. Doing this manually could take a considerable amount of time and resources and thus we use GridSearchCV to automate the tuning of hyperparameters.

GridSearchCV is a function that comes in Scikit-learn's(or SK-learn) model\_selection package. So, we need to have Scikit-learn library installed on the computer. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.

### How does GridSearchCV work?

Pass predefined values for hyperparameters to the GridSearchCV function by defining a dictionary in which a particular hyperparameter is mentioned along with the values it can take.

GridSearchCV tries all the combinations of the values passed in the dictionary and evaluates the model for each combination using the Cross-Validation method. Hence after using this function we get accuracy/loss for every combination of hyperparameters and we can choose the one with the best performance.

### Python implementation of grid search using GridSearchCV of the sklearn library

```
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVR

gsc = GridSearchCV(
    estimator=SVR(kernel='rbf'),
    param_grid={
        'C': [0.1, 1, 100, 1000],
        'epsilon': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05,
        0.1, 0.5, 1, 5, 10],
        'gamma': [0.0001, 0.001, 0.005, 0.1, 1, 3, 5]
    },
    cv=5, scoring='neg_mean_squared_error', verbose=0, n_jobs=-1)
```

First, we need to import GridSearchCV from the sklearn library, a machine learning library for python. The estimator parameter of GridSearchCV requires the model we are using for the hyper parameter tuning process. For this example, we are using the rbf kernel of the Support Vector Regression model (SVR). The param\_grid parameter requires a list of parameters and the range of values for each parameter of the specified estimator. The most significant parameters required when working with the rbf kernel of the SVR model are c, gamma and epsilon. A list of values to choose from should be given to each hyper parameter of the model. You can change these values and experiment more to see which value ranges give better performance. A cross validation process is performed in order to determine the hyper parameter value set which provides the best accuracy levels.

```
grid_result = gsc.fit(X, y)
best_params = grid_result.best_params_

best_svr = SVR(kernel='rbf', C=best_params["C"],
                 epsilon=best_params["epsilon"], gamma=best_params["gamma"],
                 coef0=0.1, shrinking=True,
                 tol=0.001, cache_size=200, verbose=False,
                 max_iter=-1)
```

## Hyperparameter optimization

Model optimization is one of the toughest challenges in the implementation of machine learning solutions. Entire branches of machine learning and deep learning theory have been dedicated to the optimization of models.

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

Hyperparameter optimization in machine learning intends to find the hyperparameters of a given machine learning algorithm that deliver the best performance as measured on a validation set. Hyperparameters, in contrast to model parameters, are set by the machine learning engineer before training. The number of trees in a random forest is a hyperparameter while the weights in a neural network are model parameters learned during training.

Some examples of model hyperparameters include:

- The learning rate for training a neural network.
- The C and  $\gamma$  hyperparameters for support vector machines.

- The k in k-nearest neighbors.

Hyperparameter optimization finds a combination of hyperparameters that returns an optimal model which reduces a predefined loss function and in turn increases the accuracy on given independent data.

Model	Overview	Hyperparameters
C4.5	J48 Decision Tree	c ={0.05,0.10,0.20,0.30,0.40,0.50,0.60,0.70}
NNET	3-layer Neural Network	size = {4,..,28}, decay = {0.10,0.20}
KNN	K- Nearest Neighbor	c ={2*(0,.....,7) + 1}
RF	Random Forest	mtry= { 10, 50,100, 200, 250,500, 1000}
SVM	Support Vector Machine	c ={ $2^{-6}, \dots, 2^{10}$ }

Image: Classification models with their respective hyperparameters

Reference: <https://www.kdnuggets.com/2020/05/hyperparameter-optimization-machine-learning-models.html>

Now, let's see the various hyperparameter optimization methods used in machine learning.

## Hyperparameter Optimization methods

Hyperparameters can have a direct impact on the training of machine learning algorithms. Thus, to achieve maximal performance, it is important to understand how to optimize them. Here are some common strategies for optimizing hyperparameters:

### 1. Manual Hyperparameter Tuning

Traditionally, hyperparameters were tuned manually by trial and error. This is still commonly done, and experienced engineers can “guess” parameter values that will deliver very high accuracy for ML models. However, there is a continual search for better, faster, and more automatic methods to optimize hyperparameters.

### 2. Grid Search

Grid search is arguably the most basic hyperparameter tuning method. With this technique, we simply build a model for each possible combination of all of the hyperparameter values provided, evaluating each model, and selecting the architecture which produces the best results.

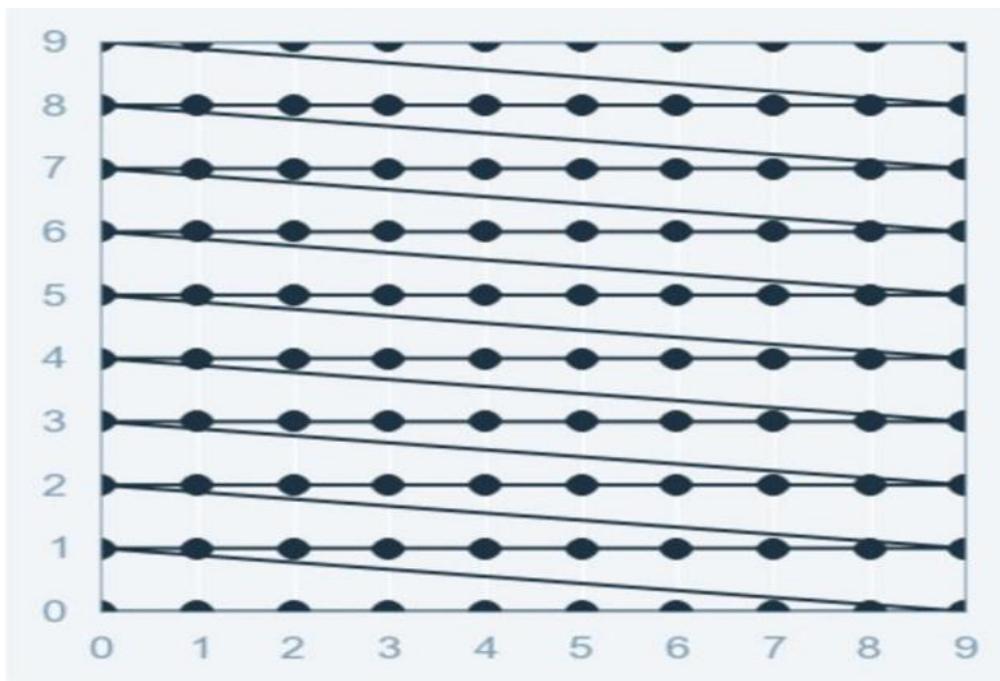


Image: Visual representation of grid search

Reference: <https://www.kdnuggets.com/2020/05/hyperparameter-optimization-machine-learning-models.html>

Grid-search does NOT only apply to one model type but can be applied across machine learning to calculate the best parameters to use for any given model. For example, a typical soft-margin SVM classifier equipped with an RBF kernel has at least two hyperparameters that need to be optimized for good performance on unseen data: a regularization constant  $C$  and a kernel hyperparameter  $\gamma$ . Both parameters are continuous, so to perform grid search, one selects a finite set of “reasonable” values for each, let’s say

$$C \in \{10, 100, 1000\}$$

$$\gamma \in \{0.1, 0.2, 0.5, 1.0\}$$

Grid search then trains an SVM with each pair  $(C, \gamma)$  in the cartesian product of these two sets and evaluates their performance on a held-out validation set (or by internal cross-validation on the training set, in which case multiple SVMs are trained per pair). Finally, the grid search algorithm outputs the settings that achieved the highest score in the validation procedure.

One of the drawbacks of grid search is that when it comes to dimensionality, it suffers when evaluating the number of hyperparameters grows exponentially. However, there is no guarantee that the search will produce the perfect solution, as it usually finds one by aliasing around the right set.

### 3. Random Search

Often some of the hyperparameters matter much more than others. Performing random search rather than grid search allows a much more precise discovery of good values for the important ones.

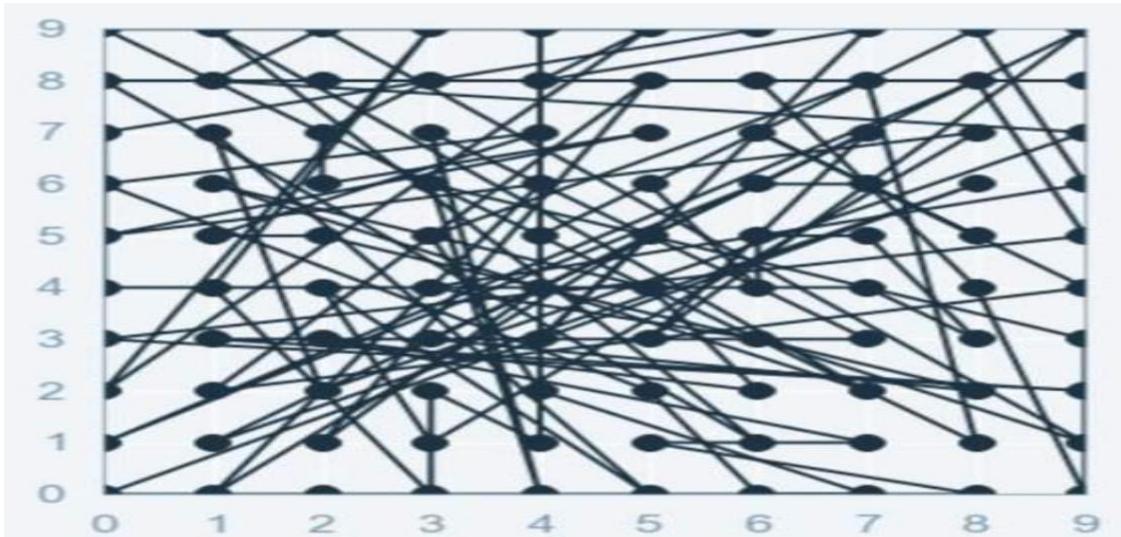


Image: Visual representation of random search

Reference: <https://www.kdnuggets.com/2020/05/hyperparameter-optimization-machine-learning-models.html>

Random Search sets up a grid of hyperparameter values and selects random combinations to train the model and score. This allows you to explicitly control the number of parameter combinations that are attempted. The number of search iterations is set based on time or resources. Scikit Learn offers the `RandomizedSearchCV` function for this process.

### 4. Bayesian Optimization

Bayesian optimization works by constructing a posterior distribution of functions (Gaussian process) that best describes the function you want to optimize. As the number of observations grows, the posterior distribution improves, and the algorithm becomes more certain of which regions in parameter space are worth exploring and which are not.

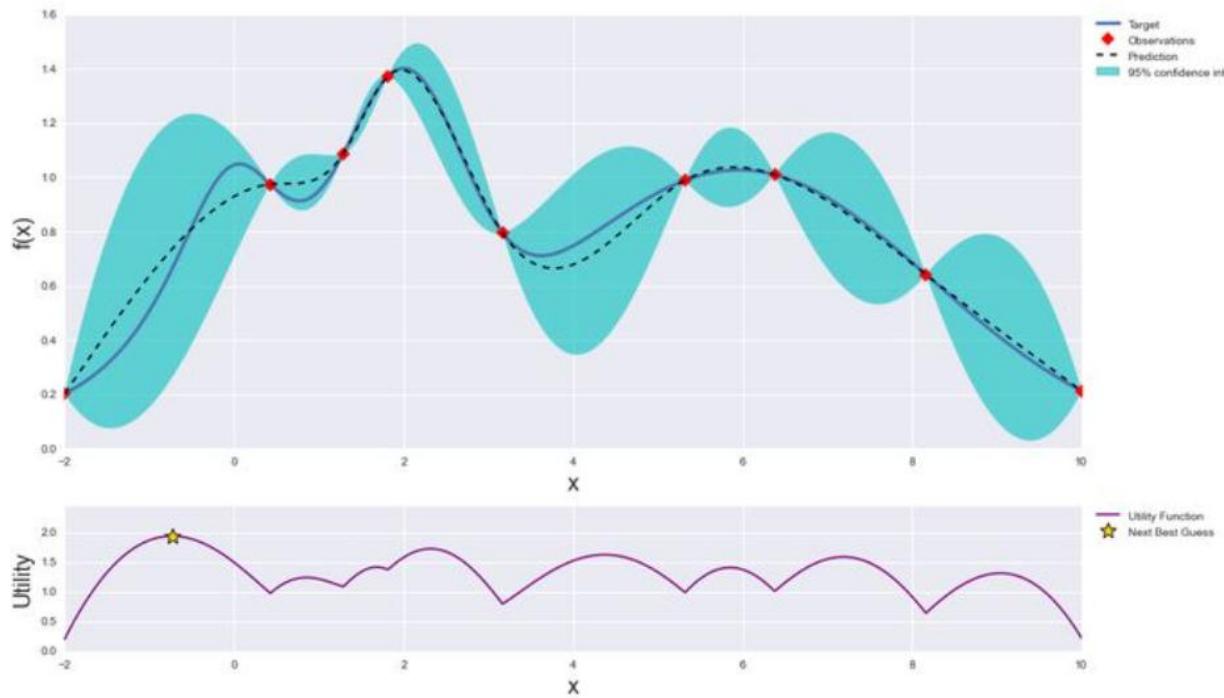


Image: Bayesian optimization

Reference: <https://www.kdnuggets.com/2020/05/hyperparameter-optimization-machine-learning-models.html>

## 5. Gradient-based Optimization

It is specially used in the case of Neural Networks. It computes the gradient with respect to hyperparameters and optimizes them using the gradient descent algorithm.

The calculation of the gradient is the least of problems. At least in times of advanced automatic differentiation software.

## 6. Evolutionary Optimization

Evolutionary optimization follows a process inspired by the biological concept of evolution and since natural evolution is a dynamic process in a changing environment, they are also well suited to dynamic optimization problems.

Evolutionary algorithms are often used to find good approximate solutions that cannot be easily solved by other techniques. Optimization problems often don't have an exact solution as it may be too time-consuming and computationally intensive to find an optimal solution. However, evolutionary algorithms are ideal in such situations as they can be used to find a near-optimal solution which is often sufficient.

One advantage of evolutionary algorithms is that they develop solutions free of any human misconceptions or biases, which means they can produce surprising ideas which we might never generate ourselves.

After exploring the concepts like how to build model, how to find best hyperparameters and various techniques of tuning let us start with another classification algorithm in machine learning named Decision tree.

## 4.13 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.

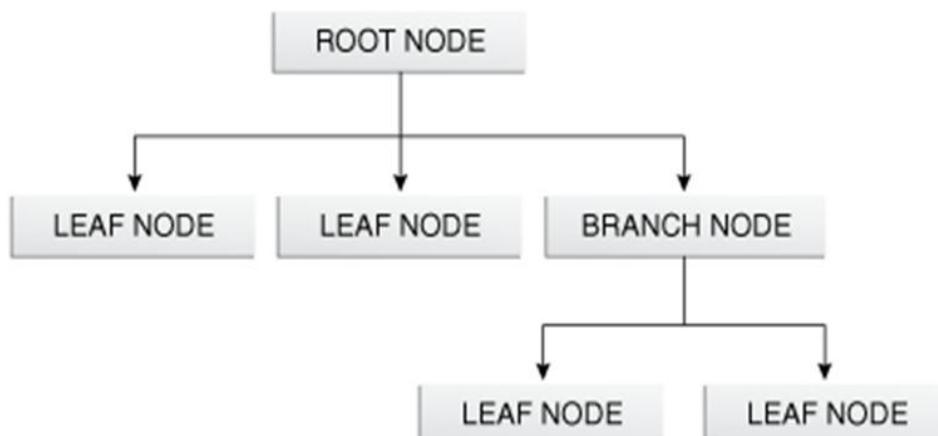


Image: General structure of a decision tree

## 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.

## Decision Tree Terminologies

1. 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.
2. Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
3. Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
4. Branch/Sub Tree: A tree formed by splitting the tree.
5. Pruning: Pruning is the process of removing the unwanted branches from the tree.
6. Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

## 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:

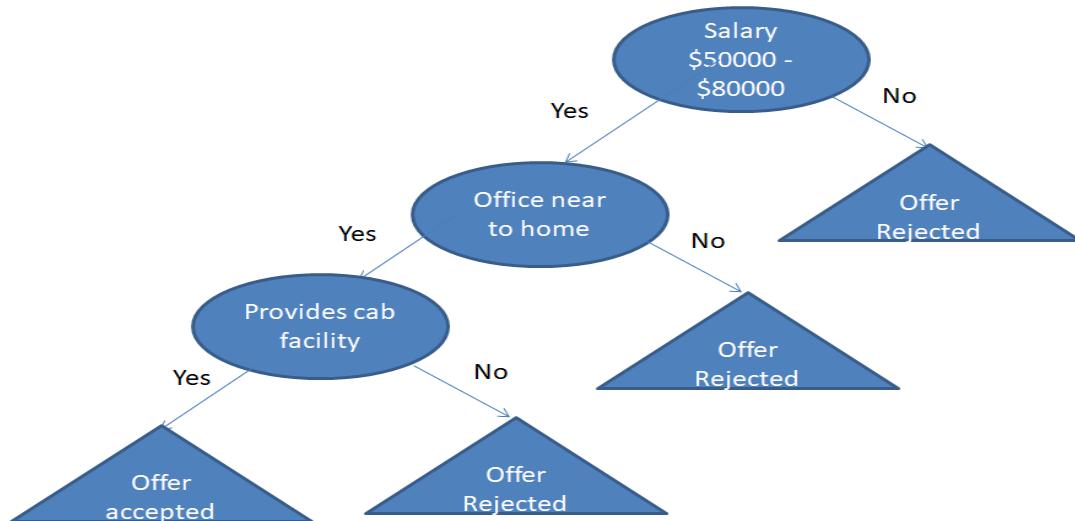


Image: Example

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.

## 4.14 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.

## Project : 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:

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/3.Decision\\_Tree.ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/3.Decision_Tree.ipynb)

Now after exploring decision tree algorithm let's get started with another very popular classification algorithm called Support Vector Machine.

## 4.15 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:

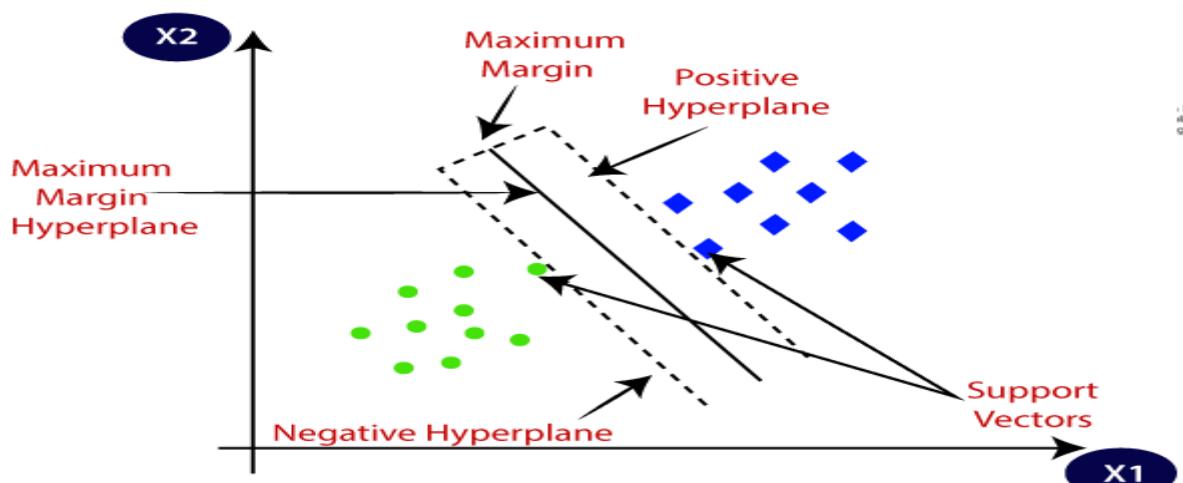


Image: Support Vector Machine

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:

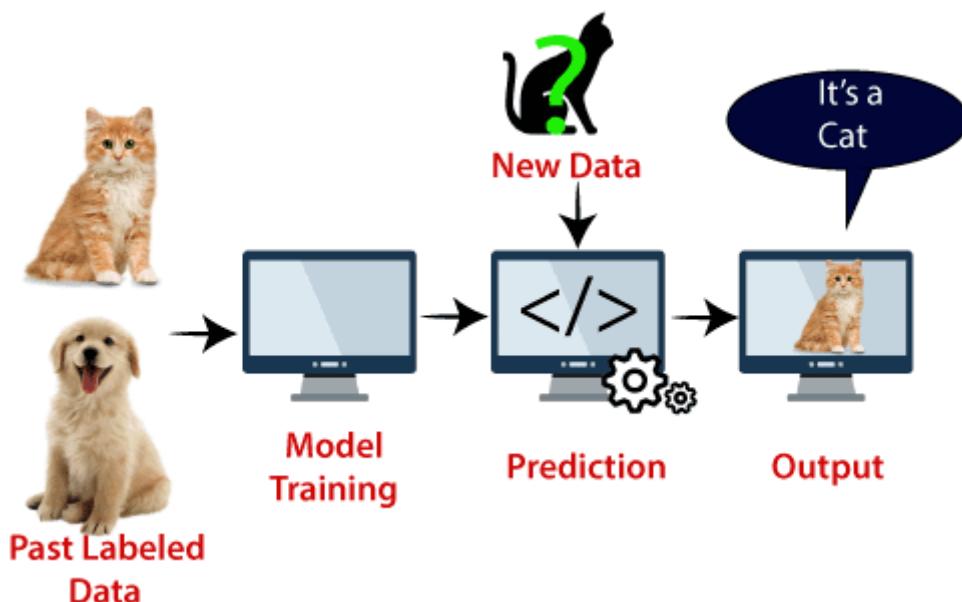


Image: Support Vector Machine Example

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

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

## Types of SVM

SVM can be of two types:

1. 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.

## 2. 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.

### 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:

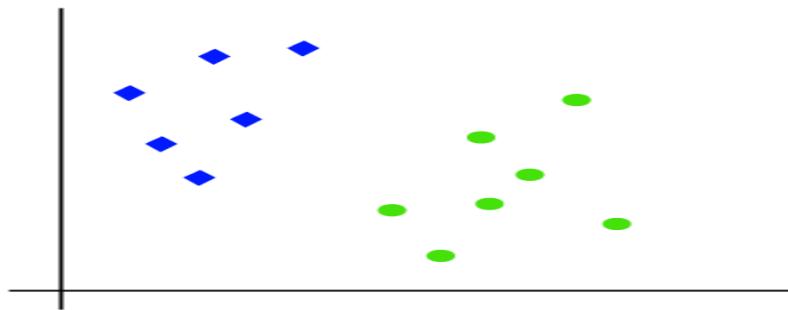


Image: Dataset with Green Blue tags

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:

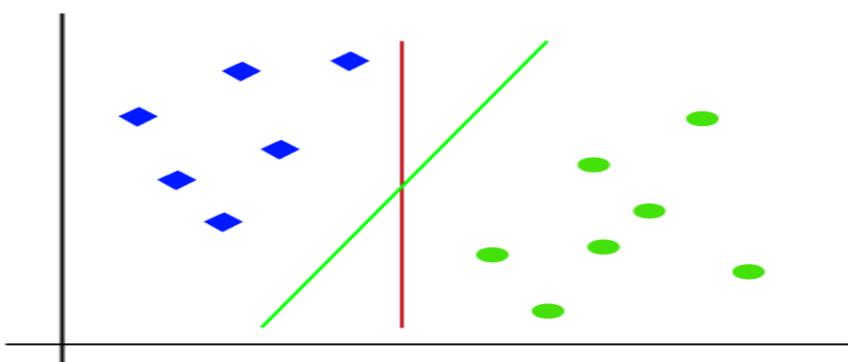


Image: Multiple lines to separate Classes

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.

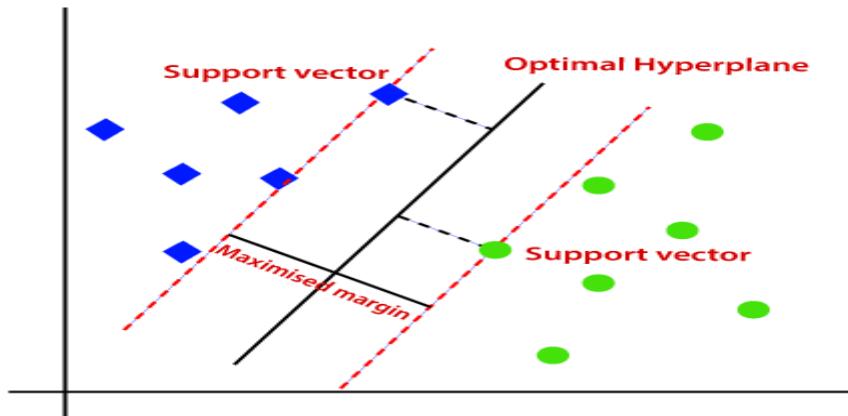


Image: Hyperplane

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.

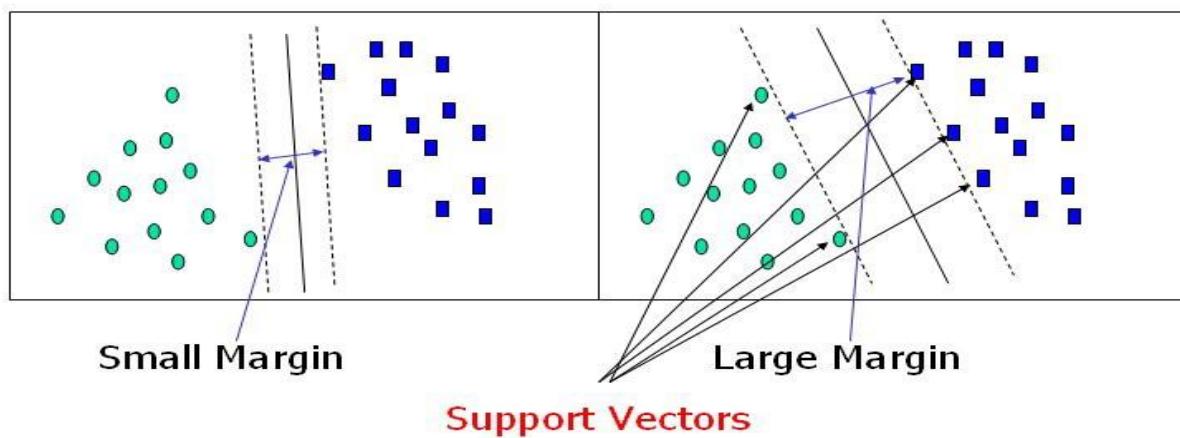


Image: Support Vectors

Reference: [https://miro.medium.com/max/700/0\\*ecA4Ls8kBYSM5nza.jpg](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.

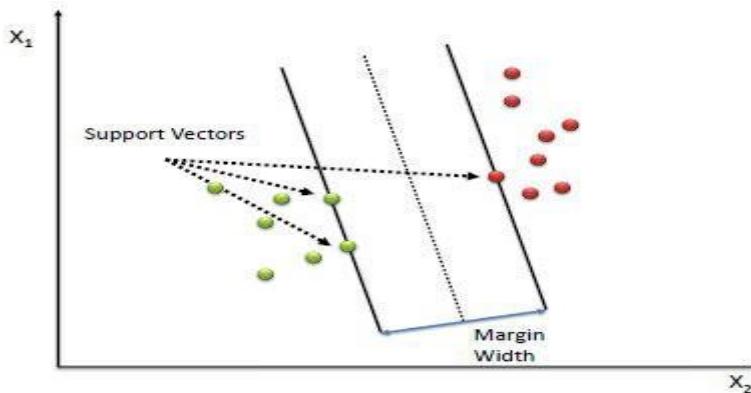


Image: Support vectors and margins

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 ( $\pi$ ,  $\pi^+$ ,  $\pi^-$ ) such that ' $\pi^+$ ' is parallel to ' $\pi$ ' passing through the support vectors on the positive side and ' $\pi^-$ ' is parallel to ' $\pi$ ' passing through the support vectors on the negative side.

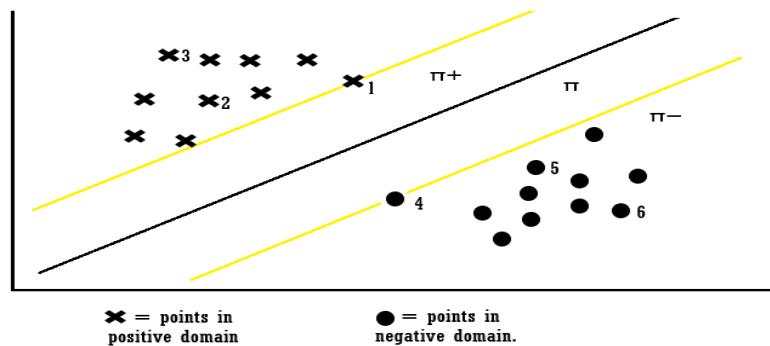


Image: Support vectors and Hyperplanes

Reference: [https://miro.medium.com/max/687/1\\*ppsJ51I8o5kTC1q2opSjwQ.png](https://miro.medium.com/max/687/1*ppsJ51I8o5kTC1q2opSjwQ.png)

the equations of each hyperplane can be considered as:

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

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:

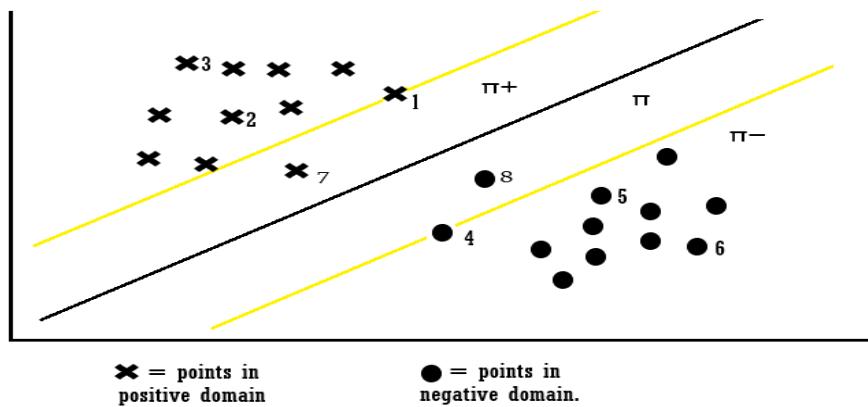


Image: Unclassified Constraints

Reference: [https://miro.medium.com/max/687/1\\*ppsJ51I8o5kTC1q2opSjwQ.png](https://miro.medium.com/max/687/1*ppsJ51I8o5kTC1q2opSjwQ.png)

for point X7:

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

**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 ( $\xi$ ) which is called  $X_i$ .

if we introduce  $\xi$  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  $\xi_i$  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;

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

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

where  $\xi_i = \zeta_i$

## Project : Linear SVM in Sklearn

Let us consider a Breast Cancer dataset. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. We will apply SVM with Linear Kernel.

Link for the project:

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20%20Notebooks/Chapter%204/4.SVM.ipynb>

After understanding the working of support vector machine algorithm let us explore about how we can calculate the distance between two datapoints and various distance calculating techniques.

## 4.16 Different distance methods

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

#### 1. 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.

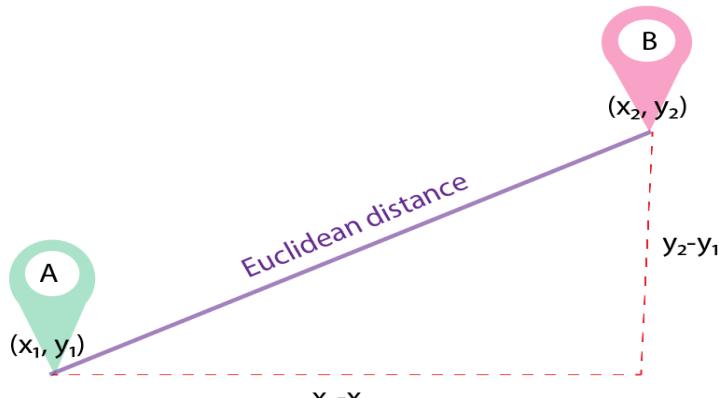


Image: Euclidean Distance

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

## 2. 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.

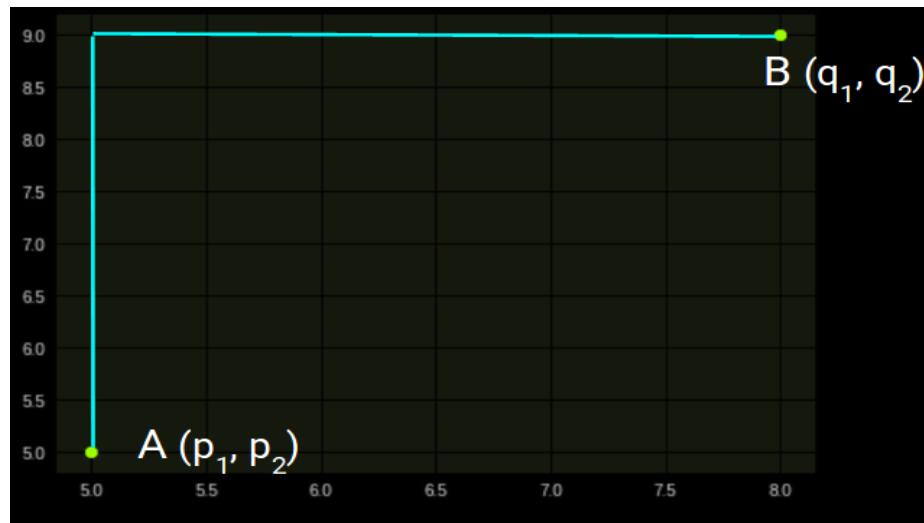


Image: Euclidean Distance between A and B

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<sub>i</sub>, q<sub>i</sub> = 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’.

### 3. 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.

### 4. 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.

## 4.17 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.



Image: KNN Classifier

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

## 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:

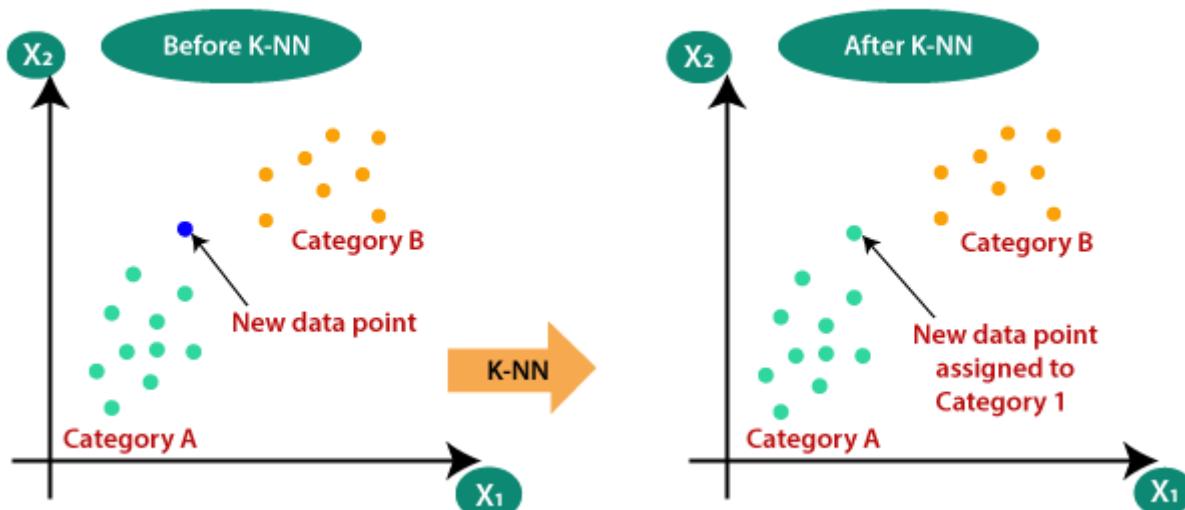


Image: Before and After KNN

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

## 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:

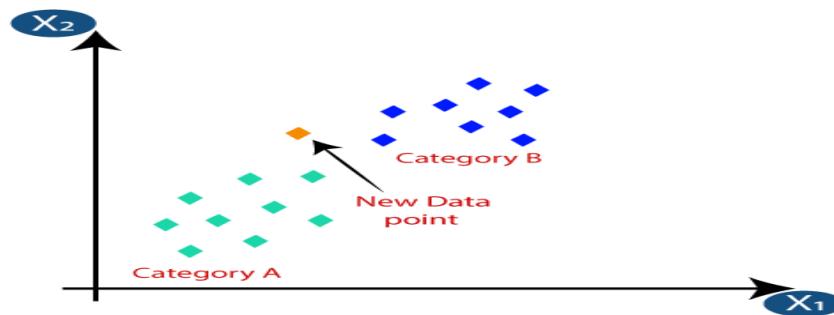


Image: New data point

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:

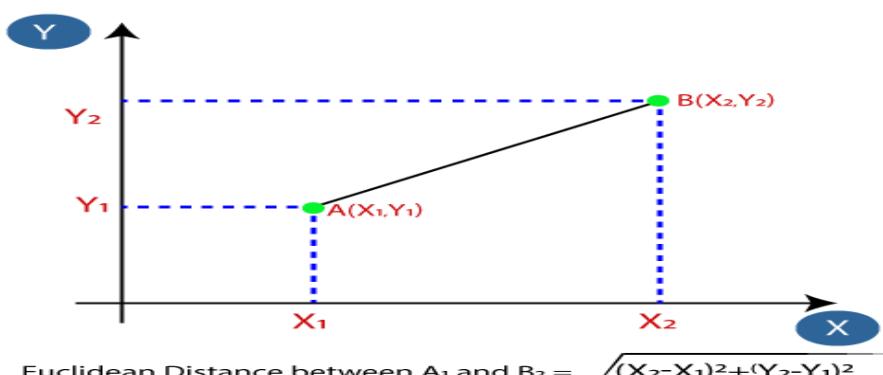


Image: Euclidean distance calculation

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:

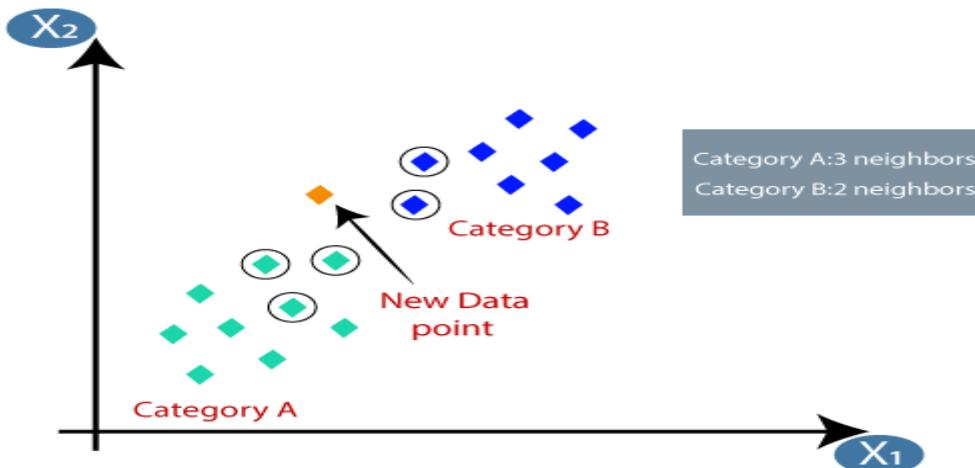


Image: Finding of Nearest neighbors

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.

## Project: KNN using Sklearn

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

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/6.Decision%20Boundary%20for%20KNN.ipynb>

We have explored the working of KNN algorithm, before starting out next algorithm lets us learn about probability concepts which are the foundations of our next Naive Bayes algorithm.

## 4.18 Probability Theory

Probability is the measure of the likelihood that an event will occur in a Random Experiment. Probability is quantified as a number between 0 and 1, where, loosely speaking, 0 indicates impossibility and 1 indicates certainty. The higher the probability of an event, the more likely it is that the event will occur.

Example:-

A simple example is the tossing of a fair (unbiased) coin. Since the coin is fair, the two outcomes (“heads” and “tails”) are both equally probable; the probability of “heads” equals the probability of “tails”; and since no other outcomes are possible, the probability of either “heads” or “tails” is 1/2 (which could also be written as 0.5 or 50%).

### **Random Experiment**

A random experiment is a physical situation whose outcome cannot be predicted until it is observed.

### **Sample Space**

A sample space, is a set of all possible outcomes of a random experiment.

Example:

Random Experiment: Toss a fair coin once.

Sample Space:  $\Omega = \{\text{Head}, \text{Tail}\}$

## Random Variable

A random variable, is a variable whose possible values are numerical outcomes of a random experiment. There are two types of random variables.

1. Discrete Random Variable is one which may take on only a countable number of distinct values such as 0,1,2,3,4,..... Discrete random variables are usually (but not necessarily) counts.
2. Continuous Random Variable is one which takes an infinite number of possible values. Continuous random variables are usually measurements.

$X$ , is defined as a function from the sample space to the real numbers.

$$X : \Omega \rightarrow \mathbb{R}$$

Example

For above random experiment

$$X = \begin{cases} 1 & \text{if Head} \\ 0 & \text{if Tail} \end{cases}$$

## Conditional Probability

Conditional Probability is a measure of the probability of an event given that (by assumption, presumption, assertion or evidence) another event has already occurred. If the event of interest is A and the event B is known or assumed to have occurred, “the conditional probability of A given B”, is usually written as  $P(A|B)$ .

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

**Example**

Suppose that somebody secretly rolls two fair six-sided dice,  
 Let D1 be the value rolled on die 1.  
 Let D2 be the value rolled on die 2.

What is the probability that D1 = 2

		D2					
		1	2	3	4	5	6
D1	1						
	2						
	3						
	4						
	5						
	6						

sample space = 36 outcomes. So D1 = 2 in exactly 6 of the 36 outcomes;  
 $P(D1=2) = 6/36 = 1/6$ .

What is the probability that  $D1+D2 \leq 5$

		D2					
		1	2	3	4	5	6
D1	1						
	2						
	3						
	4	1					
	5						
	6						

$D1+D2 \leq 5$  for exactly 10 of the same 36 outcomes, thus  $P(D1+D2 \leq 5) = 10/36$ .

What is the probability that  $D1 = 2$  given that  $D1+D2 \leq 5$

		D2					
		1	2	3	4	5	6
D1	1						
	2						
	3						
	4	1					
	5						
	6						

For  $D1 = 2$ , there are 3 out of 10 outcomes,

So the conditional probability  $P(D1=2 | D1+D2 \leq 5) = 3/10 = 0.3$

## Independence

Two events are said to be independent of each other, if the probability that one event occurs in no way affects the probability of the other event occurring, or in other words if we have observation about one event it doesn't affect the probability of the other. For Independent events A and B below is true

$$P(A, B) = P(A) * P(B) \quad \text{where} \quad P(A) \neq 0 \quad \text{and} \quad P(B) \neq 0$$

$$P(A | B) = P(A) \quad \text{and} \quad P(B | A) = P(A)$$

**Example:** -

Let's say you rolled a die and flipped a coin. The probability of getting any number face on the die is no way influences the probability of getting a head or a tail on the coin.

### Conditional Independence

Two events A and B are conditionally independent given a third event C precisely if the occurrence of A and the occurrence of B are independent events in their conditional probability distribution given C. In other words, A and B are conditionally independent given C if and only if, given knowledge that C already occurred, knowledge of whether A occurs provides no additional information on the likelihood of B occurring, and knowledge of whether B occurs provides no additional information on the likelihood of A occurring.

$$P(A | C, B | C) = P(A | C) * P(B | C) \quad \text{where} \quad P(A | C) \neq 0 \quad \text{and} \quad P(B | C) \neq 0$$

### Example: -

A box contains two coins, a regular coin and one fake two-headed coin ( $P(H)=1$ ,  $P(T)=0$ ). I choose a coin at random and toss it twice.

Let

A = First coin toss results in an HH.

B = Second coin toss results in an HH.

C = Coin 1 (regular) has been selected.

If C is already observed i.e., we already know whether a regular coin is selected or not, the event A and B becomes independent as the outcome of 1 doesn't affect the outcome of another event.

### Expectation

The expectation of a random variable X is written as  $E(X)$ . If we observe N random values of X, then the mean of the N values will be approximately equal to  $E(X)$  for large N. In more concrete terms, the expectation is what you would expect the outcome of an experiment to be on an average if you repeat the experiment a large number of time.

If X is a continuous random variable with p.d.f.  $F_X(x)$

$$E[X] = \int_{-\infty}^{\infty} x F_X(x) dx$$

If X is a discrete random variable with probability function  $F_X(x)$ .

$$E[X] = \sum_x x F_X(x) = \sum_x x P(X = x)$$

If  $f(X)$  is a function of X then the Expected value of  $f(X)$

$$E[f(X)] = \sum_x f(x) F_X(x) = \sum_x f(x) P(X = x)$$

**Example**

Expected value when we roll a fair die (random experiment).

Let X represent the outcome of the experiment.

$X = \{1, 2, 3, 4, 5, 6\}$ , Each of these has a probability of 1/6 of occurring as it's a fair die.

So

$$\begin{aligned} E[X] &= 1 * 1/6 + 2 * 1/6 + 3 * 1/6 + 4 * 1/6 + 5 * 1/6 + 6 * 1/6 \\ &= 7/2 = 3.5 \end{aligned}$$

So the expectation is 3.5 . If you think about it, 3.5 is halfway between the possible values the die can take and so this is what you should have expected.

## Variance

The variance of a random variable X is a measure of how concentrated the distribution of a random variable X is around its mean. It's defined as

$$Var[X] = E[X - E[X]^2] = E[X^2] - [E[X]]^2$$

**Example**

For the above experiment with die the variance is.

Let  $f(X) = X^2$ , Then using the last definition of Expectation

$$E[X^2] = E[f(X)] = \sum_x f(x) P(X = x)$$

$$E[X^2] = f(1)P(x = 1) + f(2)P(x = 2) + \dots + f(6)P(x = 6) = 15.6$$

$$Var[X] = E[X^2] - (E[X])^2 = 15.6 - 3.5^2$$

$$= 3.35$$

## Probability Distribution

Is a mathematical function that maps the all possible outcomes of an random experiment with it's associated probability. It depends on the Random Variable X, whether it's discrete or continues.

1. Discrete Probability Distribution: The mathematical definition of a discrete probability function,  $p(x)$ , is a function that satisfies the following properties. This is referred as Probability Mass Function.

1. The probability that  $x$  can take a specific value is  $p(x)$ , i.e.

$$P[X=x]=p(x)$$

2.  $p(x)$  is non-negative for all real  $x$ .

3. The sum of  $p(x)$  over all possible values of  $x$  is 1, i.e.

$$\sum_x P(x) = 1$$

Example: For single coin flip the PMF can be represented in a table as below

X	Head	Tail
P(x)		
	0.5	0.5

2. Continuous Probability Distribution: The mathematical definition of a continuous probability function,  $f(x)$ , is a function that satisfies the following properties. This is referred as Probability Density Function.

1. The probability that  $x$  is in between two points  $a$  and  $b$  is  $P(x)$

$$P[a \leq x \leq b] = \int_a^b f(x)dx$$

2.  $f(x)$  is non-negative for all real  $x$ .

3. The sum of  $p(x)$  over all possible values of  $x$  is 1, i.e.

$$\int_{-\infty}^{\infty} f(x)dx = 1$$

## 4.19 Naïve Bayes Classifier Algorithm

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

- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

## Why is it called Naïve Bayes?

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

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

## Bayes' Theorem

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

The formula for Bayes' theorem is given as:

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

where,

- $P(A|B)$  is Posterior probability: Probability of hypothesis A on the observed event B.
- $P(B|A)$  is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.
- $P(A)$  is Prior Probability: Probability of hypothesis before observing the evidence.
- $P(B)$  is Marginal Probability: Probability of Evidence.

## Advantages of Naïve Bayes Classifier

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems.

## Disadvantages of Naïve Bayes Classifier

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

## Applications of Naïve Bayes Classifier

- It is used for Credit Scoring.
- It is used in medical data classification.
- It can be used in real-time predictions because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as Spam filtering and Sentiment analysis.

## Types of Naïve Bayes Model

There are three types of Naive Bayes Model, which are given below:

1. Gaussian: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
2. Multinomial: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc.
3. Bernoulli: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

## Project : Naïve Bayes using Sklearn

Let us consider a dataset about sports. We have 2 features telling us about weather and humidity. The target what we need to find is whether a person will play or not . Let us use GaussianNB to predict the output.

Link for the project : [https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/7.Naive\\_Bayes.py](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/7.Naive_Bayes.py)

## Project: Text Analytics using Pandas

Text Analytics is the process of drawing meaning out of written communication. Let us see how to perform Text Analytics using Pandas library

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/8.Text\\_Analytics\\_with\\_Pandas.ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/8.Text_Analytics_with_Pandas.ipynb)

### 4.20 Bag of Words Approach

The bag-of-words model is a way of representing text data when modeling text with machine learning algorithms. The bag-of-words model is simple to understand and implement and has seen great success in problems such as language modeling and document classification.

A bag-of-words model, or BoW for short, is a way of extracting features from text for use in modeling, such as with machine learning algorithms. The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents. A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:

- A vocabulary of known words.
- A measure of the presence of known words.

It is called a “*bag*” of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.

Here's a sample of reviews about a particular horror movie.

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

We can see that there are some contrasting reviews about the movie as well as the length and pace of the movie. Imagine looking at a thousand reviews like these. Clearly, there is a lot of interesting insights we can draw from them and build upon them to gauge how well the movie performed.

However, as we saw above, we cannot simply give these sentences to a machine learning model and ask it to tell us whether a review was positive or negative. We need to perform certain text preprocessing steps.

We will first build a vocabulary from all the unique words in the above three reviews. The vocabulary consists of these 11 words: 'This', 'movie', 'is', 'very', 'scary', 'and', 'long', 'not', 'slow', 'spooky', 'good'. We can now take each of these words and mark their occurrence in the three movie reviews above with 1s and 0s. This will give us 3 vectors for 3 reviews:

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

Image: Bag of Words Approach

Reference: <https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/BoWBag-of-Words-model-2.png>

Vector of Review 1: [1 1 1 1 1 1 1 0 0 0 0]

Vector of Review 2: [1 1 2 0 0 1 1 0 1 0 0]

Vector of Review 3: [1 1 1 0 0 0 1 0 0 1 1]

And that's the core idea behind a Bag of Words (BoW) model.

## Term Frequency-Inverse Document Frequency (TF-IDF)

Term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

### *Term Frequency (TF)*

***It is a measure of how frequently a term, t, appears in a document, d:***

$$tf_{t,d} = \frac{n_{t,d}}{\text{Number of terms in the document}}$$

Here, in the numerator, n is the number of times the term “t” appears in the document “d”. Thus, each document and term would have its own TF value. Take the same vocabulary we had built in the Bag-of-Words model to show how to calculate the TF for Review #2:

Review 2: This movie is not scary and is slow

Here,

- Vocabulary: ‘This’, ‘movie’, ‘is’, ‘very’, ‘scary’, ‘and’, ‘long’, ‘not’, ‘slow’, ‘spooky’, ‘good’
- Number of words in Review 2 = 8
- TF for the word ‘this’ = (number of times ‘this’ appears in review 2)/ (number of terms in review 2) = 1/8

Similarly,

- TF(‘movie’) = 1/8
- TF(‘is’) = 2/8 = 1/4
- TF(‘very’) = 0/8 = 0
- TF(‘scary’) = 1/8
- TF(‘and’) = 1/8
- TF(‘long’) = 0/8 = 0
- TF(‘not’) = 1/8
- TF(‘slow’) = 1/8
- TF(‘spooky’) = 0/8 = 0
- TF(‘good’) = 0/8 = 0

We can calculate the term frequencies for all the terms and all the reviews in this manner:

Term	Review 1	Review 2	Review 3	TF (Review 1)	TF (Review 2)	TF (Review 3)
This	1	1	1	1/7	1/8	1/6
movie	1	1	1	1/7	1/8	1/6
is	1	2	1	1/7	1/4	1/6
very	1	0	0	1/7	0	0
scary	1	1	0	1/7	1/8	0
and	1	1	1	1/7	1/8	1/6
long	1	0	0	1/7	0	0
not	0	1	0	0	1/8	0
slow	0	1	0	0	1/8	0
spooky	0	0	1	0	0	1/6
good	0	0	1	0	0	1/6

Image: Term Frequency (TF)

Reference: <https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/TF-matrix-1.png>

### Inverse Document Frequency (IDF)

IDF is a measure of how important a term is. We need the IDF value because computing just the TF alone is not sufficient to understand the importance of words.

$$idf_t = \log \frac{\text{number of documents}}{\text{number of documents with term } 't'}$$

We can calculate the IDF values for all the words in Review 2:

$$\text{IDF('this')} = \log \left( \frac{\text{number of documents}}{\text{number of documents containing the word 'this'}} \right) = \log \left( \frac{3}{3} \right) = \log (1) = 0$$

Similarly,

- $\text{IDF}(\text{'movie'}) = \log \left( \frac{3}{3} \right) = 0$
- $\text{IDF}(\text{'is'}) = \log \left( \frac{3}{3} \right) = 0$
- $\text{IDF}(\text{'not'}) = \log \left( \frac{3}{1} \right) = \log (3) = 0.48$
- $\text{IDF}(\text{'scary'}) = \log \left( \frac{3}{2} \right) = 0.18$
- $\text{IDF}(\text{'and'}) = \log \left( \frac{3}{3} \right) = 0$
- $\text{IDF}(\text{'slow'}) = \log \left( \frac{3}{1} \right) = 0.48$

We can calculate the IDF values for each word like this. Thus, the IDF values for the entire vocabulary would be:

Term	Review 1	Review 2	Review 3	IDF
This	1	1	1	0.00
movie	1	1	1	0.00
is	1	2	1	0.00
very	1	0	0	0.48
scary	1	1	0	0.18
and	1	1	1	0.00
long	1	0	0	0.48
not	0	1	0	0.48
slow	0	1	0	0.48
spooky	0	0	1	0.48
good	0	0	1	0.48

Image: IDF values for each word

Reference: <https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/IDF-matrix.png>

Hence, we see that words like “is”, “this”, “and”, etc., are reduced to 0 and have little importance; while words like “scary”, “long”, “good”, etc. are words with more importance and thus have a higher value.

We can now compute the TF-IDF score for each word in the corpus. Words with a higher score are more important, and those with a lower score are less important:

$$(tf\_idf)_{t,d} = tf_{t,d} * idf_t$$

We can now calculate the TF-IDF score for every word in Review 2:

$$\text{TF-IDF}(\text{'this'}, \text{Review 2}) = \text{TF}(\text{'this'}, \text{Review 2}) * \text{IDF}(\text{'this'}) = 1/8 * 0 = 0$$

Similarly,

- $\text{TF-IDF}(\text{'movie'}, \text{Review 2}) = 1/8 * 0 = 0$
- $\text{TF-IDF}(\text{'is'}, \text{Review 2}) = 1/4 * 0 = 0$
- $\text{TF-IDF}(\text{'not'}, \text{Review 2}) = 1/8 * 0.48 = 0.06$
- $\text{TF-IDF}(\text{'scary'}, \text{Review 2}) = 1/8 * 0.18 = 0.023$
- $\text{TF-IDF}(\text{'and'}, \text{Review 2}) = 1/8 * 0 = 0$
- $\text{TF-IDF}(\text{'slow'}, \text{Review 2}) = 1/8 * 0.48 = 0.06$

Similarly, we can calculate the TF-IDF scores for all the words with respect to all the reviews:

Term	Review 1	Review 2	Review 3	IDF	TF-IDF (Review 1)	TF-IDF (Review 2)	TF-IDF (Review 3)
This	1	1	1	0.00	0.000	0.000	0.000
movie	1	1	1	0.00	0.000	0.000	0.000
is	1	2	1	0.00	0.000	0.000	0.000
very	1	0	0	0.48	0.068	0.000	0.000
scary	1	1	0	0.18	0.025	0.022	0.000
and	1	1	1	0.00	0.000	0.000	0.000
long	1	0	0	0.48	0.068	0.000	0.000
not	0	1	0	0.48	0.000	0.060	0.000
slow	0	1	0	0.48	0.000	0.060	0.000
spooky	0	0	1	0.48	0.000	0.000	0.080
good	0	0	1	0.48	0.000	0.000	0.080

Image: TF-IDF scores

Reference: [https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/TF\\_IDF-matrix.png](https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/TF_IDF-matrix.png)

We have now obtained the TF-IDF scores for our vocabulary. TF-IDF also gives larger values for less frequent words and is high when both IDF and TF values are high i.e. the word is rare in all the documents combined but frequent in a single document.

### Project for TFIDF using Sklearn:

We will work on using the TfidfVectorizer to learn vocabulary and inverse document frequencies across 2 small documents and then encode one of those documents. A vocabulary of multiple words is learned from the documents and each word is assigned a unique integer index in the output vector.

Link:

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/9.Bag%20of%20Word%20with%20Vectorization%20Technique\(TFIDF\).ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/9.Bag%20of%20Word%20with%20Vectorization%20Technique(TFIDF).ipynb)

### Project on Spam Ham Using using BOW concept:

Spam Ham classification is one of the classical application of Machine Learning. There are two types of data present in this repository, which is **ham** (non-spam)

and **spam** data. Furthermore, in the ham data, there are easy and hard, which means there is some non-spam data that has a very high similarity with spam data. This might pose a difficulty for our system to make a decision. Let us use the concept of Bag of Words to convert Text to Numbers and then apply Classification algorithm on it.

Link for the project:

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/10.%20Spam\\_Ham\\_Demonstration.ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/10.%20Spam_Ham_Demonstration.ipynb)

## 4.21 Lexicons for Sentiment Analysis

Sentiment analysis is a process by which information is analyzed through the use of natural language processing (NLP). Lexicons calculate the sentiment from the semantic orientation of word or phrases that occur in a text.

The types of lexicons are:

- Affin
- Textblob
- VADER

### Afinn

It is the simplest yet popular lexicons used for sentiment analysis developed by Finn Årup Nielsen. It contains 3300+ words with a polarity score associated with each word. In python, there is an in-built function for this lexicon. [10.6]

### Textblob

It is a simple python library that offers API access to different NLP tasks such as sentiment analysis, spelling correction, etc. Textblob sentiment analyzer returns two properties for a given input sentence:

- Polarity is a float that lies between [-1,1], -1 indicates negative sentiment and +1 indicates positive sentiments.
- Subjectivity is also a float which lies in the range of [0,1]. Subjective sentences generally refer to personal opinion, emotion, or judgment.

## VADER sentiment

Valence aware dictionary for sentiment reasoning (VADER) is another popular rule-based sentiment analyzer. It uses a list of lexical features (e.g. word) which are labeled as positive or negative according to their semantic orientation to calculate the text sentiment. Vader sentiment returns the probability of a given input sentence to be Positive, negative, and neutral.

For example:

“The food was great!”

Positive: 99%

Negative :1%

Neutral: 0%

These three probabilities will add up to 100%.

## Project : Sentiment Analysis using Lexicons

We will use all the 3 lexicons mentioned in the following project

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/11.Sentiment%20Analysis%20using%20Affin%20Lexicon.ipynb>

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/12.Sentiment%20Analysis%20using%20Textblob%20Lexicon.ipynb>

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%204/13.Sentiment%20Analysis%20using%20Vader%20Lexicon.ipynb>

## 4.22 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. [11.1]

## 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.

## 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.

## 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.

### 4.23 Clustering

It is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labelled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data

points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

For ex– The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.[11.4]

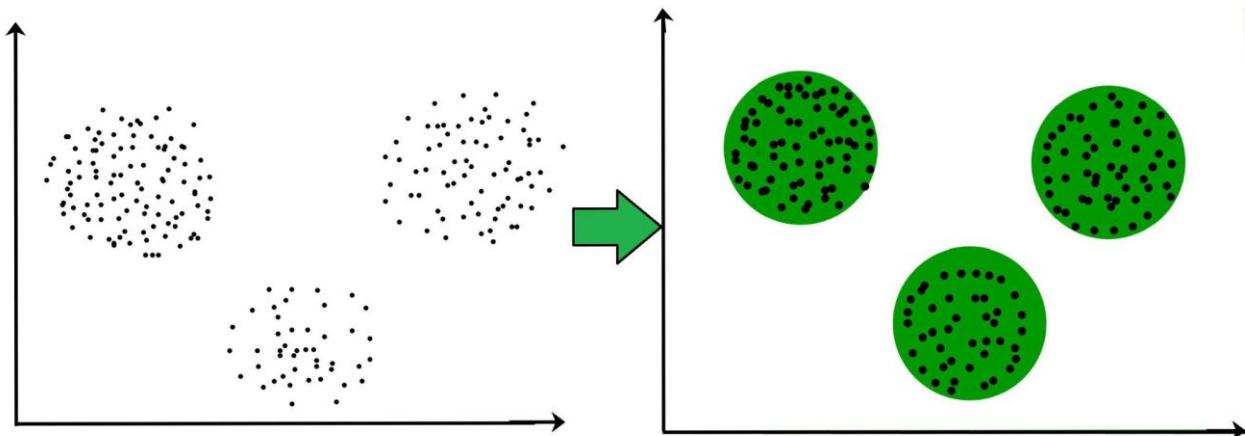


Fig : Clustering

Reference:<https://www.geeksforgeeks.org/clustering-in-machine-learning/#:~:text=Clustering%20is%20the%20task%20of,data%20points%20in%20other%20groups.>

It is not necessary for clusters to be a spherical. Such as :

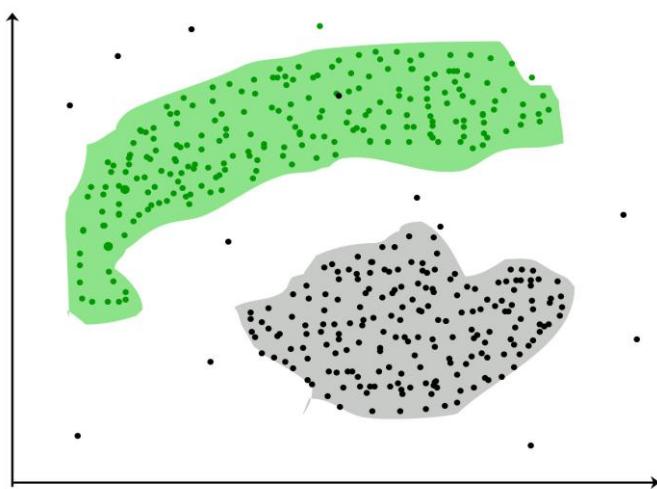


Fig : Unspespherical Clustering

Reference : <https://www.geeksforgeeks.org/clustering-in-machine-learning/#:~:text=Clustering%20is%20the%20task%20of,data%20points%20in%20other%20groups.>

## Why Clustering ?

Clustering is very much important as it determines the intrinsic grouping among the unlabelled data present. There are no criteria for a good clustering. It depends on the user, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding “natural clusters” and describe their unknown properties (“natural” data types), in finding useful and suitable groupings (“useful” data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions which constitute the similarity of points and each assumption make different and equally valid clusters.

## Clustering Methods :

- **Density-Based Methods:** These methods consider the clusters as the dense region having some similarity and different from the lower dense region of the space. These methods have good accuracy and ability to merge two clusters. Example *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)* , *OPTICS (Ordering Points to Identify Clustering Structure)* etc.
- **Hierarchical Based Methods:** The clusters formed in this method forms a tree-type structure based on the hierarchy. New clusters are formed using the previously formed one. It is divided into two category Agglomerative (*bottom up approach*) and Divisive (*top down approach*). Some examples: *CURE (Clustering Using Representatives)*, *BIRCH (Balanced Iterative Reducing Clustering and using Hierarchies)* etc.
- **Partitioning Methods:** These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter. For e.g. *K-means*, *CLARANS (Clustering Large Applications based upon Randomized Search)* etc.
- **Grid-based Methods:** In this method the data space is formulated into a finite number of cells that form a grid-like structure. All the clustering operation done on these grids are fast and independent of the number of data objects example *STING (Statistical Information Grid)*, *wave cluster*, *CLIQUE (CLustering In Quest)* etc.

### 4.24 Agglomerative Clustering

Agglomerative clustering refers to a collection of clustering algorithms that all build upon the same principles: the algorithm starts by declaring each point its own cluster, and then merges the two most similar clusters until some stopping criterion is satisfied. The stopping criterion implemented in scikit-learn is the number of clusters,

so similar clusters are merged until only the specified number of clusters are left. There are several linkage criteria that specify how exactly the “most similar cluster” is measured. This measure is always defined between two existing clusters.

The following three choices are implemented in scikit-learn:

- Ward - The default choice, ward picks the two clusters to merge such that the variance within all clusters increases the least. This often leads to clusters that are relatively equally sized.
- Average - average linkage merges the two clusters that have the smallest average distance between all their points.
- Complete - complete linkage (also known as maximum linkage) merges the two clusters that have the smallest maximum distance between their points.

The following figure illustrates the progression of agglomerative clustering on a two-dimensional dataset, looking for three clusters:

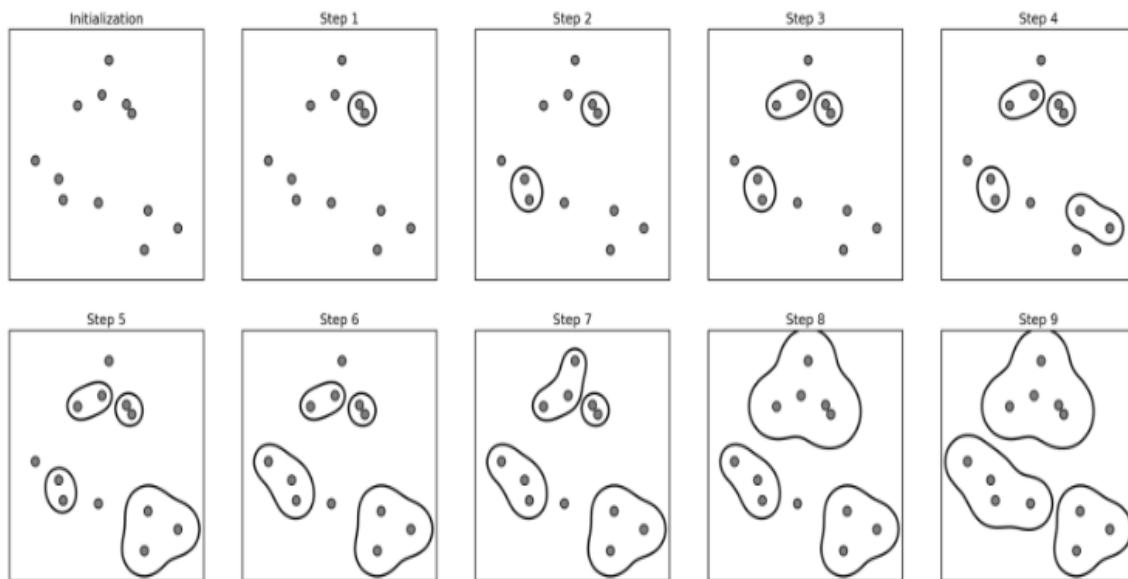


Fig : Process of Agglomerative Clustering

Reference : Andreas C. Müller and Sarah Guido , Introduction to Machine learning with Python , O'reilly , October 2016

After exploring various methods of clustering and Agglomerative clustering technique lets understand next algorithm DBSCAN.

## 4.25 DBSCAN

Another very useful clustering algorithm is DBSCAN (which stands for “density based spatial clustering of applications with noise”). The main benefits of DBSCAN are that it does not require the user to set the number of clusters a priori, it can

capture clusters of complex shapes, and it can identify points that are not part of any cluster. DBSCAN is somewhat slower than agglomerative clustering and k-means, but still scales to relatively large datasets.

DBSCAN works by identifying points that are in “crowded” regions of the feature space, where many data points are close together. These regions are referred to as dense regions in feature space. The idea behind DBSCAN is that clusters form dense regions of data, separated by regions that are relatively empty.

Points that are within a dense region are called core samples (or core points), and they are defined as follows.

There are two parameters in DBSCAN: min\_samples and eps. If there are at least min\_samples data points within a distance of eps to a given data point, that data point is classified as a core sample. Core samples that are closer to each other than the distance eps are put into the same cluster by DBSCAN.

The algorithm works by picking an arbitrary point to start with. It then finds all points with distance eps or less from that point. If there are less points than min\_samples points within distance eps of the starting point, this point is labelled as noise, meaning that it doesn't belong to any cluster. If there are more points than min\_samples points within a distance of eps , the point is labelled a core sample and assigned a new cluster label.

Then, all neighbours (within eps) of the point are visited. If they have not been assigned a cluster yet, they are assigned the new cluster label that was just created. If they are core samples, their neighbours are visited in turn, and so on. The cluster grows until there are no more core samples within distance eps of the cluster. Then another point that hasn't yet been visited is picked, and the same procedure is repeated.

In the end, there are three kinds of points: core points, points that are within distance eps of core points (called boundary points), and noise. When the DBSCAN algorithm is run on a particular dataset multiple times, the clustering of the core points is always the same, and the same points will always be labelled as noise. However, a boundary point might be neighbour to core samples of more than one cluster. Therefore, the cluster membership of boundary points depends on the order in which points are visited. Usually there are only few boundary points, and this slight dependence on the order of points is not important.

## 4.26 K-means clustering

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.

A cluster refers to a collection of data points aggregated together because of certain similarities. You'll define a target number  $k$ , which refers to the number of centroids you need in the dataset. A centroid is the imaginary or real location representing the center of the cluster. Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares. In other words, the K-means algorithm identifies  $k$  number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The '*means*' in the K-means refers to averaging of the data; that is, finding the centroid.

### **How the K-means algorithm works**

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids.

It halts creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.

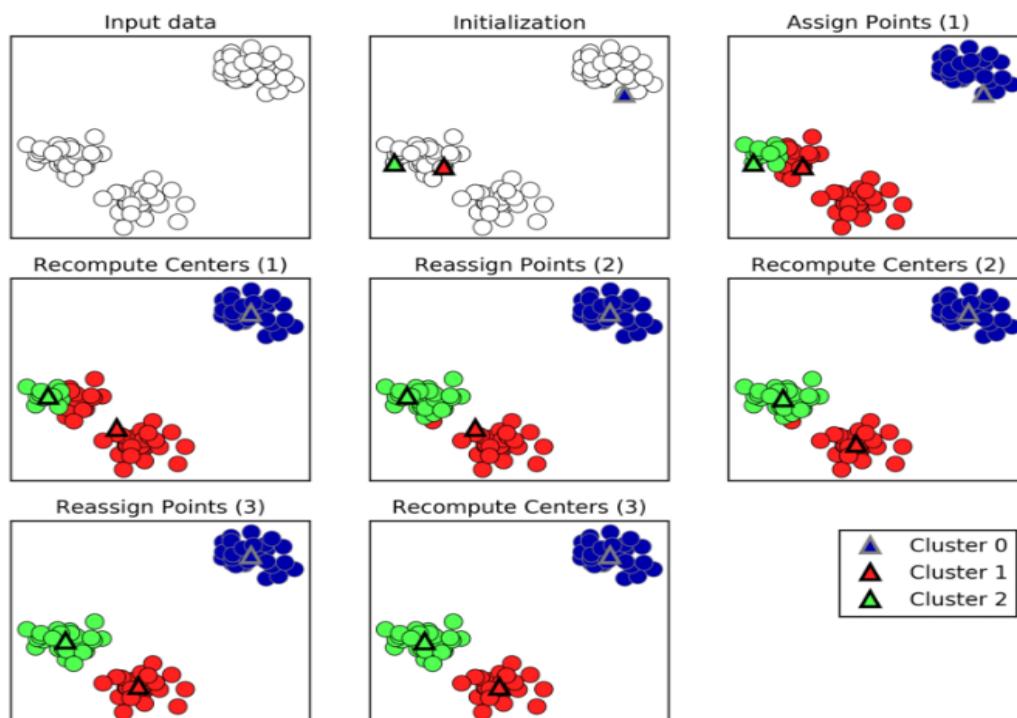


Fig : k-means clustering steps

Reference - Andreas C. Müller and Sarah Guido , Introduction to Machine learning with Python , O'reilly , October 2016.

Please refer to the link below for a nice animated view of k-means clustering.

[https://commons.wikimedia.org/wiki/File:K-means\\_convergence.gif](https://commons.wikimedia.org/wiki/File:K-means_convergence.gif)

## Project: K-Means Clustering

Using the Iris Dataset, we will work on building an unsupervised learning based on 2 features of the dataset.

Link for the project code:

[https://github.com/Edunet-Foundation/Tech-Saksham/tree/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/1.k means clustering](https://github.com/Edunet-Foundation/Tech-Saksham/tree/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/1.k%20means%20clustering.ipynb)

We will also work on another 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.

Link for the project

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/2.Customer%20Segmentation/Pract Customer%20Segmentation.ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/2.Customer%20Segmentation/Pract%20Customer%20Segmentation.ipynb)

So, in this chapter we explored the various algorithms which falls under the category of supervised and unsupervised machine learning. But there are some drawbacks or limitations on these algorithms. So, in the next chapter we are going to discuss the subset of machine learning named deep learning and how it helps to overcome the limitations of machine learning. So let us get started with deep learning.

# Chapter 5: Building Deep Learning Models

## Learning Outcomes:

- Understand basic key concepts of Deep Learning
- Understand the process of automatic feature extraction using Artificial Neural Networks
- Understand the basics of Convolutional Neural Networks
- Use Transfer Learning for building Convolutional Neural Networks
- Demonstrate OCR using Tesseract software
- Able to design ANN, CNN and transfer learning models

## 5.1 Deep Learning Basics

Deep learning (also known as deep structured learning) is a part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. Deep learning models are not meant to be trained with an algorithm. Instead, they make learning a step further. Deep learning models works directly with audio, images and video data to get real time analysis. The data being fed to the deep learning model does not need any external intervention. You can feed raw data to the model and receive actionable insights.

Deep-learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, machine vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and

symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue.

The adjective "deep" in deep learning refers to the use of multiple layers in the network. Early work showed that a linear perceptron cannot be a universal classifier, and then that a network with a nonpolynomial activation function with one hidden layer of unbounded width can on the other hand so be. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability, whence the "structured" part.

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level *on its own*. (Of course, this does not completely eliminate the need for hand-tuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction.)

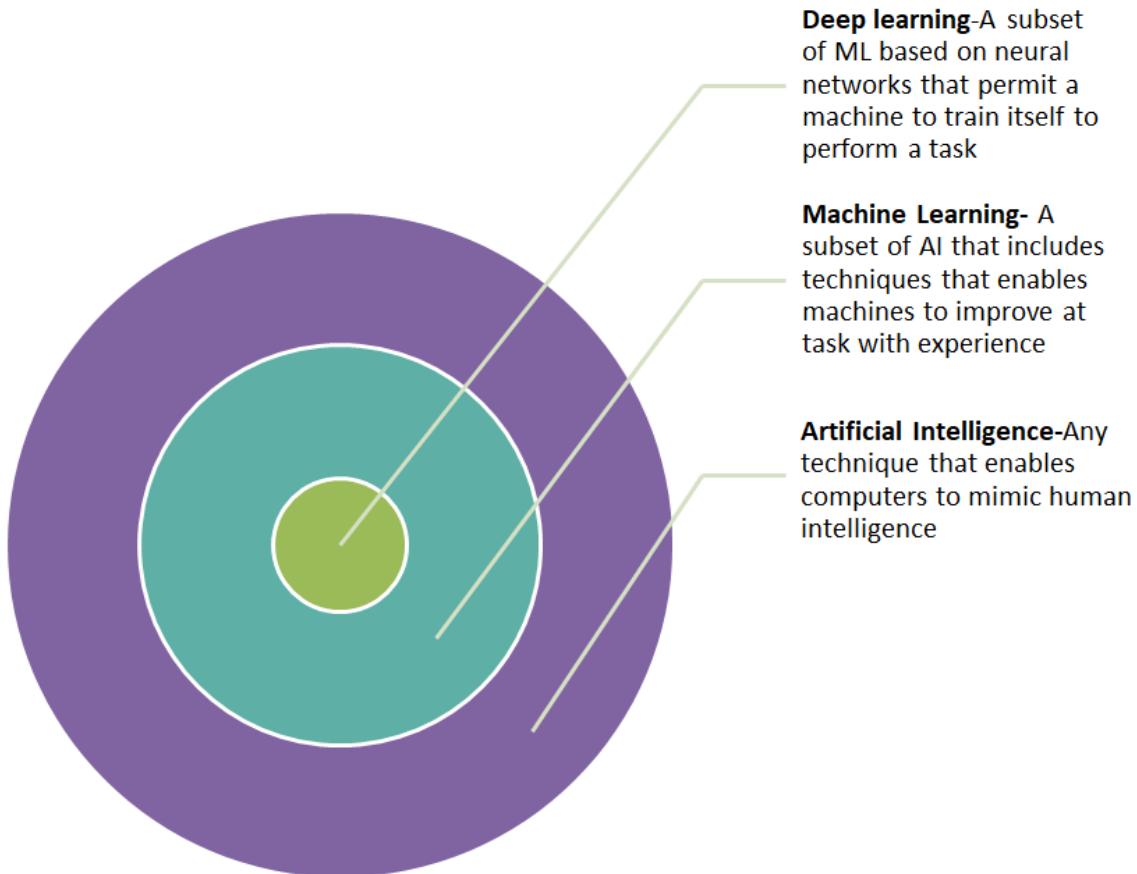


Fig : Relation between Deep Learning , Machine Learning and Artificial Intelligence

## Differences between Machine Learning and Deep Learning

Machine Learning	Deep Learning
Outputs: Numerical Value, like classification of score	Anything from numerical values to free-form elements, such as free text and sound.
Uses various types of automated algorithms that turn to model functions and predict future action from data.	Uses neural network that passes data through processing layers to the interpret data features and relations.
Algorithms are detected by data analysts to examine specific variables in data sets.	Algorithms are largely self-depicted on data analysis once they're put into production.

Now let's explore the building blocks of neural networks which are mainly inspired by working of human brain.

## 5.2 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.

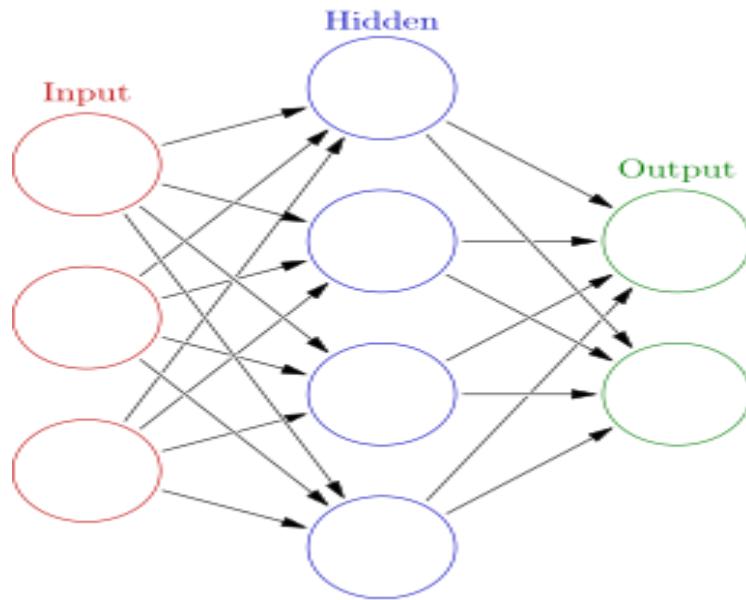


Fig : Neural Network

Reference - Glosser.ca, CC BY-SA 3.0 <<https://creativecommons.org/licenses/by-sa/3.0>>, via Wikimedia Commons

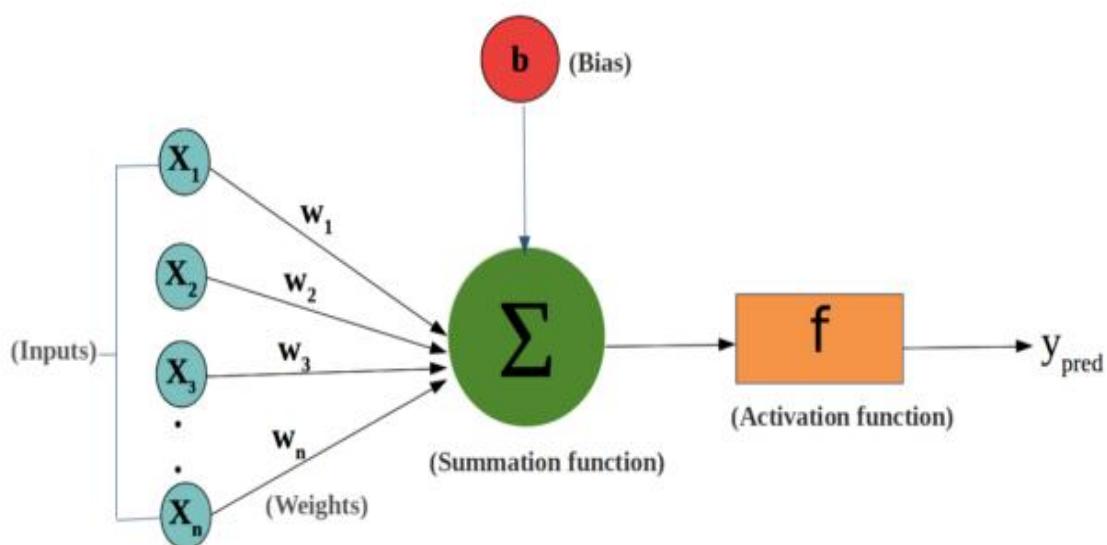


Fig : Basic building blocks of Neural Networks

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.

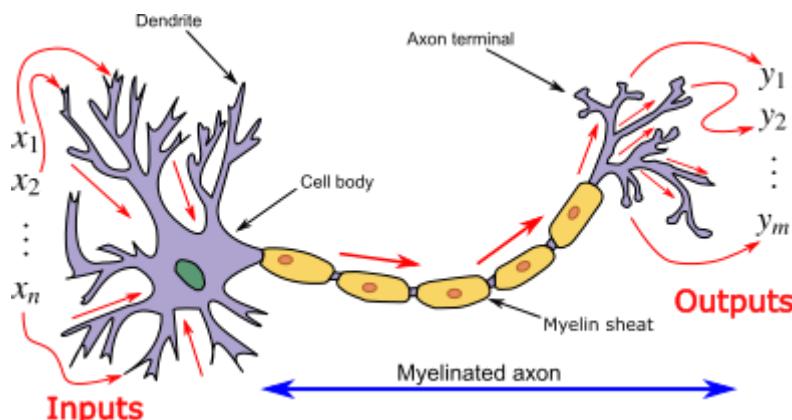


Fig : Neuron

Reference - Egm4313.s12 at English Wikipedia, CC BY-SA 3.0 <<https://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons

The activation function of a node defines the output of that node given an input or set of inputs.

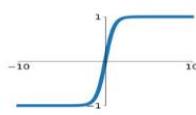
### Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



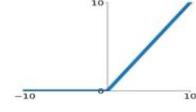
### tanh

$$\tanh(x)$$



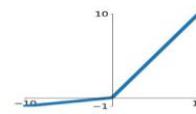
### ReLU

$$\max(0, x)$$



### Leaky ReLU

$$\max(0.1x, x)$$



### Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

### ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

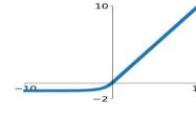


Fig : Different Activation functions

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.

## 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.

## 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.

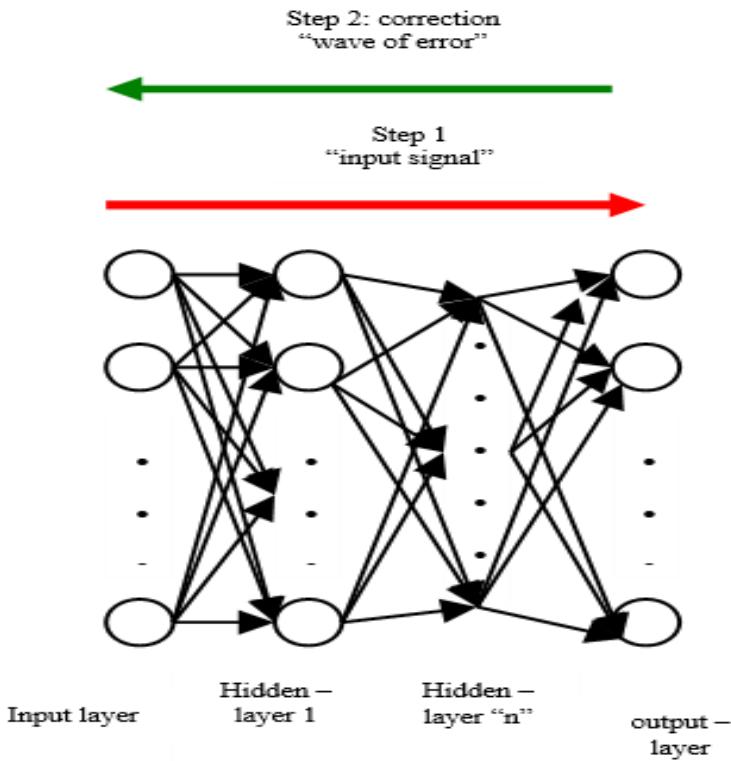


Fig : Forward and backward propagation in Neural Networks  
 Reference - Jorge Guerra Pires, CC BY-SA 3.0 <<https://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons

After understanding the concepts like neuron, bias, weights, forward propagation, backward propagation which plays a vital role in predicting final output. Let's explore some problems like overfitting and underfitting of a model.

## Project on Gradient Descent Implementation

We have understood how Gradient Descent works theoretically. Let us implement this practically in Python

Link

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/3.Gradient\\_Descent/Pract\\_Gradient\\_Descent.ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/3.Gradient_Descent/Pract_Gradient_Descent.ipynb)

## 5.3 Overfitting in Deep Learning

Overfitting is "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably". An overfitted model is a statistical model that contains more

parameters than can be justified by the data. The essence of overfitting is to have unknowingly extracted some of the residual variation (i.e. the noise) as if that variation represented underlying model structure.

Underfitting occurs when a statistical model cannot adequately capture the underlying structure of the data. An under-fitted model is a model where some parameters or terms that would appear in a correctly specified model are missing. Under-fitting would occur, for example, when fitting a linear model to non-linear data. Such a model will tend to have poor predictive performance.

The possibility of over-fitting exists because the criterion used for selecting the model is not the same as the criterion used to judge the suitability of a model. For example, a model might be selected by maximizing its performance on some set of training data, and yet its suitability might be determined by its ability to perform well on unseen data; then overfitting occurs when a model begins to "memorize" training data rather than "learning" to generalize from a trend.

As an extreme example, if the number of parameters is the same as or greater than the number of observations, then a model can perfectly predict the training data simply by memorizing the data in its entirety. Such a model, though, will typically fail severely when making predictions.

The potential for overfitting depends not only on the number of parameters and data but also the conformability of the model structure with the data shape, and the magnitude of model error compared to the expected level of noise or error in the data. Even when the fitted model does not have an excessive number of parameters, it is to be expected that the fitted relationship will appear to perform less well on a new data set than on the data set used for fitting (a phenomenon sometimes known as *shrinkage*). In particular, the value of the coefficient of determination will shrink relative to the original data.

To lessen the chance of, or amount of, overfitting, several techniques are available (e.g. model comparison, cross-validation, regularization, early stopping, pruning, Bayesian priors, or dropout). The basis of some techniques is either (1) to explicitly penalize overly complex models or (2) to test the model's ability to generalize by evaluating its performance on a set of data not used for training, which is assumed to approximate the typical unseen data that a model will encounter.

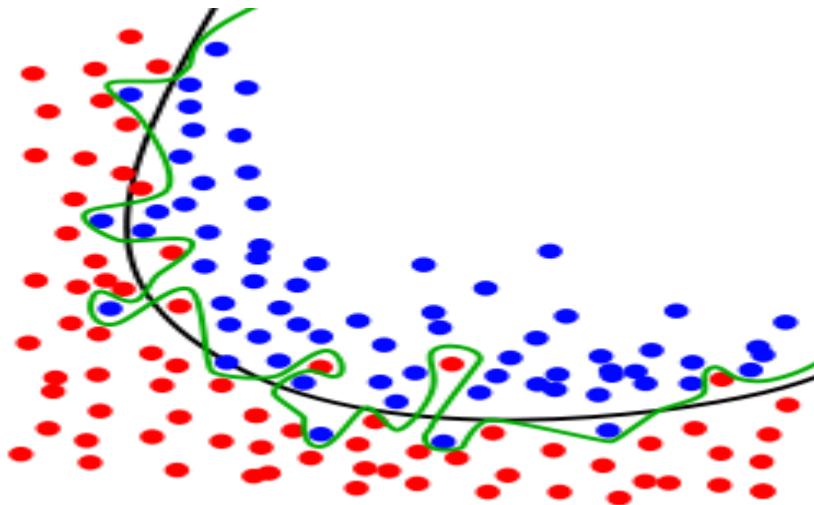


Fig : Green line shows Overfitting

Reference - Chabacano, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0>>, via Wikimedia Commons

## Measures to prevent overfitting

### *Reducing network complexity*

Deep neural networks like CNN are prone to overfitting because of the millions or billions of parameters it encloses. A model with these many parameters can overfit on the training data because it has sufficient capacity to do so.

By removing certain layers or decreasing the number of neurons (filters in CNN) the network becomes less prone to overfitting as the neurons contributing to overfitting are removed or deactivated. The network also has a reduced number of parameters because of which it cannot memorize all the data points & will be forced to generalize.

There is no general rule as to how many layers are to be removed or how many neurons must be in a layer before the network can overfit. The popular approach for reducing the network complexity is

- Grid search can be applied to find out the number of neurons and/or layers to reduce or remove overfitting.
- The overfit model can be pruned (trimmed) by removing nodes or connections until it reaches suitable performance on test data.

### *Data Augmentation*

One of the best strategies to avoid overfitting is to increase the size of the training dataset. As discussed, when the size of the training data is small the network tends to have greater control over the training data. But in real-world scenarios gathering of large amounts of data is a tedious & time-consuming task, hence the collection of new data is not a viable option.

Data augmentation provides techniques to increase the size of existing training data without any external addition. If our training data consists of images, image augmentation techniques like rotation, horizontal & vertical flipping, translation, increasing or decreasing the brightness or adding noise, cutouts etc can be applied to the existing training images to increase the number of instances.

By applying the above-mentioned data augmentation strategies, the network is trained on multiple instances of the same class of object in different perspectives.

The concept behind data augmentation is that by increasing the size of the training dataset, the network is unable to overfit on all the input samples (original images + augmented images) & thus is forced to generalize. The overall training loss increases because the network doesn't predict accurately on the augmented images thus increasing the training loss & the optimizer (optimization algorithm) tunes the network to capture the generalized trend in the training data.

### *Weight Regularization*

Weight regularization is a technique which aims to stabilize an overfitted network by penalizing the large value of weights in the network. An overfitted network usually presents with problems with a large value of weights as a small change in the input can lead to large changes in the output. For instance, when the network is given new or test data, it results in incorrect predictions.

Weight regularization penalizes the network's large weights & forcing the optimization algorithm to reduce the larger weight values to smaller weights, and this leads to stability of the network & presents good performance. In weight regularization, the network configuration remains unchanged only modifying the value of weights.

### *Dropouts*

Dropout is a regularization strategy that prevents deep neural networks from overfitting. While L1 & L2 regularization reduces overfitting by modifying the loss function, dropouts, on the other hand, deactivate a certain number of neurons at a layer from firing during training.

At each iteration different set of neurons are deactivated & this results in a different set of results. Many deep learning frameworks implement dropouts as a layer which receives inputs from the previous layer, the dropout layer randomly selects neurons which are not fired to the next layer. By deactivating certain neurons which might contribute to overfitting the performance of the network on test data improves.

Dropouts reduce overfitting in a variety of problems like image classification, image segmentation, word embedding etc.

### *Importance of validation split*

The definitions of training, validation, and test sets can be fairly nuanced, and the terms are sometimes inconsistently used. In the deep learning community, “test-time inference” is often used to refer to evaluation on data in production, which is not the technical definition of a test set.

One of the most likely culprits for this disconnect between results in development vs results in production is a poorly chosen validation set (or even worse, no validation set at all). Depending on the nature of your data, choosing a validation set can be the most important step. Although sklearn offers a `train_test_split` method, this method takes a random subset of the data, which is a poor choice for many real-world problems.

When creating a machine learning model, the ultimate goal is for it to be accurate on new data, not just the data you are using to build it. Consider the below example of 3 different models for a set of data:

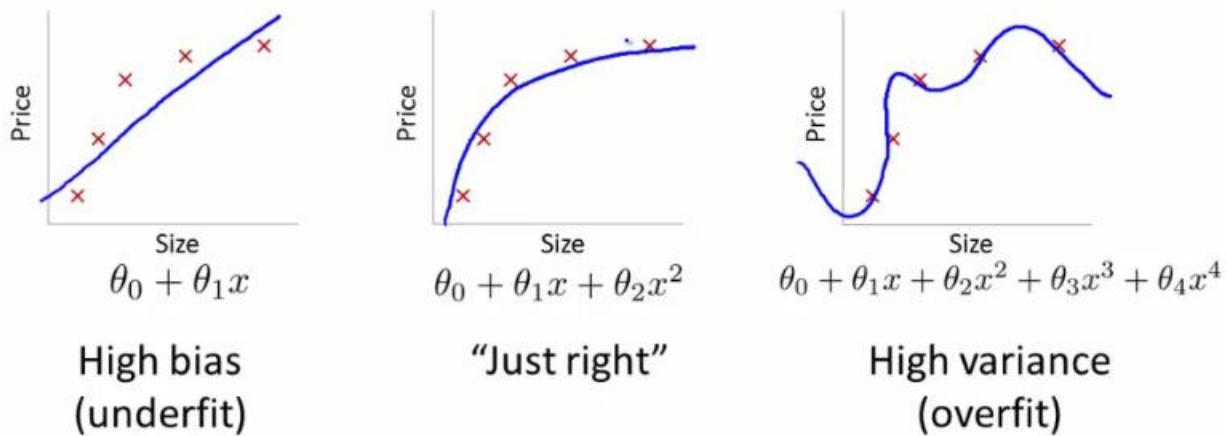


Fig : Bias Variance Tradeoff

Reference - <https://www.quora.com/What-is-an-intuitive-explanation-for-bias-variance-tradeoff>

The error for the pictured data points is lowest for the model on the far right (the blue curve passes through the red points almost perfectly), yet it's not the best choice. Why is that? If you were to gather some new data points, they most likely would not be on that curve in the graph on the right, but would be closer to the curve in the middle graph.

The underlying idea is that:

- the training set is used to train a given model
- the validation set is used to choose between models (for instance, does a random forest or a neural net work better for your problem? do you want a random forest with 40 trees or 50 trees?)

- the test set tells you how you've done. If you've tried out a lot of different models, you may get one that does well on your validation set just by chance, and having a test set helps make sure that is not the case.

A key property of the validation and test sets is that they must be representative of the new data you will see in the future. This may sound like an impossible order! By definition, you haven't seen this data yet. But there are still a few things you know about it.[13.3]

We frequently take insights from validation errors to tune our models. So we are implicitly leaking information from our validation data to our model. Advanced validation methods have obscured the importance of single split validation data. K-fold cross-validation is quite robust and probably the current industry standard for model performance validation and parameter tuning. So if you are using cross-validation techniques in your analysis, you may ignore the validation data split.

The primary objective of test data is to give an unbiased estimate of model accuracy. It should be used at the very end and only for a couple of times. If you tune your model after looking at the test accuracies, you are technically leaking information and hence cheating.

For the very same reason as above (*leakage of information*), in spite of the programming convenience we should not combine train-validation-test dataset to make common preprocessing flow. Some might argue that according to the base hypothesis train-validation-test data come from the same population distribution and hence there should be no harm in combining them for a common preprocessing flow. This is true in idealistic scenarios, but real life is far from it as you never know when your real-time production system starts getting evolving data (*whose distribution is slightly different from the training data*). As a good Data Scientist you should strive to make a model flow that is generalizable and performs well (*without any additional changes*) *irrespective* of the uncertainties in future data.

We are supposed to develop 2 separate pre-processing pipelines. (A) for training data and (B) for validation and test data. However, it should be noted that these pipelines aren't completely independent. You learn the transformation features (*mean/range/standard-deviation*) from training data and use it to transform your validation and test data. [13.4]

## Practical Code – Loss Function (MSE and Cross Entropy)

In this section, we will implement Loss Functions from scratch . For Regression we will use Mean Square Error and for classification we will use Cross Entropy.

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/4.Cross\\_Entropy\\_Vs\\_MSE/Pract\\_Cross\\_Entropy\\_MSE.ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/4.Cross_Entropy_Vs_MSE/Pract_Cross_Entropy_MSE.ipynb)

## Practical Code – Tensors in Deep Learning

Tensors form the base for creating Neural Networks. Let us see some of the popular ways of creating tensors and playing around with them.

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/5.Tensors/Pract\\_Tensors.ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/5.Tensors/Pract_Tensors.ipynb)

## Practical Code – Different Types of Layers in Keras

There are many layers which are available in Keras. We have Flatten layer for converting the values from 2D to 1 D. Next, we have Dense Layer which are regularly densely connected layers

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/6.Dense\\_Flatten\\_Layers/Pract\\_Dense\\_Flatten\\_Layers.ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/6.Dense_Flatten_Layers/Pract_Dense_Flatten_Layers.ipynb)

## Practical Code – Regularization in Deep Learning

The simplest and perhaps most common regularization method is to add a penalty to the loss function in proportion to the size of the weights in the model. A callback is an object that can perform actions at various stages of training (e.g. at the start or end of an epoch, before or after a single batch, etc).

Practical Code:

[https://github.com/Edunet-Foundation/Tech-Saksham/tree/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/8.Regularization\\_Techniques](https://github.com/Edunet-Foundation/Tech-Saksham/tree/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/8.Regularization_Techniques)

[https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/9.Custom\\_Callbacks/Pract\\_Custom\\_Callbacks.ipynb](https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/9.Custom_Callbacks/Pract_Custom_Callbacks.ipynb)

Now let us start with the basics of computer vision.

## 5.4 Computer Vision Basics

### What is Computer Vision?

Computer Vision is a field of study that helps computer understand the content of digital images such as videos and Photographs. Smartphones have cameras, and taking a photo or video and sharing it has never been easier, resulting in the incredible growth of modern social networks like Instagram.

The internet is comprised of text and images. It is relatively straightforward to index and search text, but in order to index and search images, algorithms need to know what the images contain. To get the most out of image data, we need computers to “see” an image and understand the content.

It is a multidisciplinary field that could broadly be called a subfield of artificial intelligence and machine learning, which may involve the use of specialized methods and make use of general learning algorithms. The goal of computer vision is to extract useful information from images.

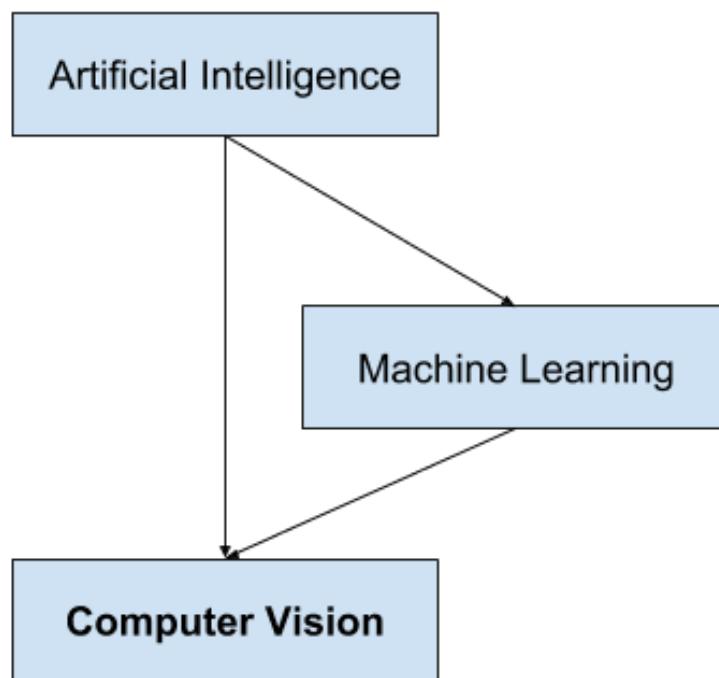


Fig : Overview of the Relationship of Artificial Intelligence and Computer Vision  
<https://machinelearningmastery.com/wp-content/uploads/2019/03/Computer-Vision.png>

Many popular computer vision applications involve trying to recognize some high-level problems; for example:

1. Image Classification
2. Object Detection
3. Optical Character Recognition
4. Image Segmentation

## Python Libraries for Computer Vision

- **OpenCV** (Open Source Computer Vision Library: <http://opencv.org>) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms.
- **Scikit-Image** - A collection of algorithms for image processing in Python.
- **SimpleCV** - An open source computer vision framework that gives access to several high-powered computer vision libraries, such as OpenCV. Written on Python and runs on Mac, Windows, and Ubuntu Linux.
- **face\_recognition** - Face recognition library that recognize and manipulate faces from Python or from the command line.
- **pytesseract** - Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images. Python-tesseract is a wrapper for Google's Tesseract-OCR Engine.

## What are Pixels?

Pixel or Picture Element is smallest addressable element in an image. Each pixel is a sample of an original image; more samples typically provide more accurate representations of the original. The intensity of each pixel is variable. *Pixels* can be used as a unit of measure such as: 2400 pixels per inch, 640 pixels per line, or spaced 10 pixels apart.

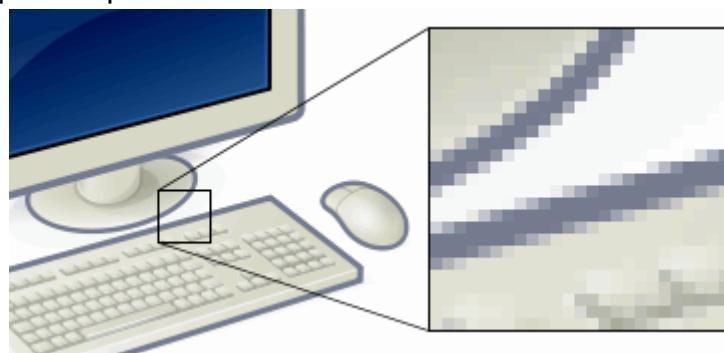


Fig : Image enlarged showing the pixels, rendered as small squares  
<https://upload.wikimedia.org/wikipedia/commons/2/2b/Pixel-example.png>

The more pixels used to represent an image, the closer the result can resemble the original. The number of pixels in an image is sometimes called the resolution, though resolution has a more specific definition. Pixel counts can be expressed as a single number, as in a "three-megapixel" digital camera, which has a nominal three million pixels, or as a pair of numbers, as in a "640 by 480 display", which has 640 pixels from side to side and 480 from top to bottom (as in a VGA display) and therefore has a total number of  $640 \times 480 = 307,200$  pixels, or 0.3 megapixels.

## Image as Matrix

Images are represented in rows and columns. For example digital grayscale image is presented in the computer by pixels matrix. Each pixel of such image is presented by one matrix element – integer from the set . The numeric values in pixel presentation are uniformly changed from zero (black pixels) to 255 (white pixels).



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29	
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0	
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1	
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49	
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36	
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62	
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0	
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0	
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19	
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0	
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0	
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4	
0	22	206	252	246	251	241	100	24	118	255	245	255	194	9	0	
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0	
0	218	251	250	137	7	11	0	0	2	62	255	250	125	3	0	
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0	
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4	
0	18	146	250	255	247	255	255	249	255	240	255	129	0	5	0	
0	0	23	113	215	255	250	248	255	248	248	118	14	12	0	0	
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1	
0	0	5	5	0	0	0	0	14	1	0	6	6	6	0	0	

Fig : Image with Pixels  
<https://mozanunal.com/images/pixel.png>

## Types of Images

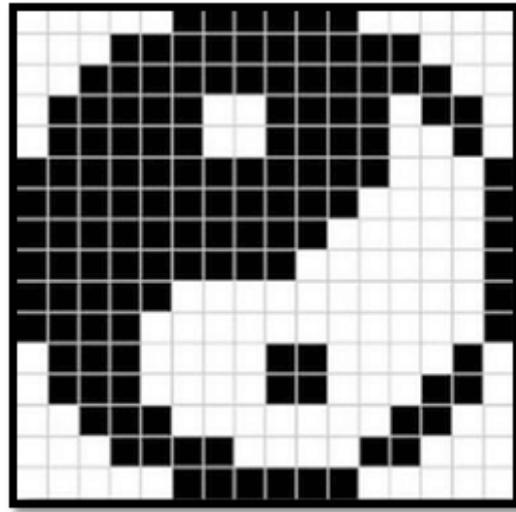
Images can be divided in 3 different categories

### 1. Binary Images

It is the simplest type of image. It takes only two values i.e, Black and White or 0 and 1. The binary image consists of a 1-bit image and it takes only 1 binary digit to represent a pixel. Binary images are mostly used for general shape or outline.

**For Example:** Optical Character Recognition (OCR).

Binary images are generated using threshold operation. When a pixel is above the threshold value, then it is turned white('1') and which are below the threshold value then they are turned black('0')



**Fig. Binary image**

<https://static.javatpoint.com/tutorial/dip/images/binary-images.png>

## 2. 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.



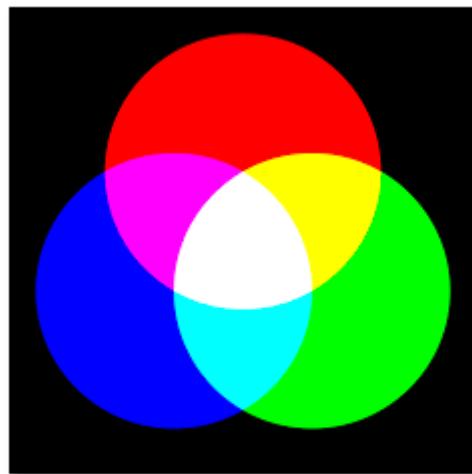
**Fig. Gray-scale image**

<https://static.javatpoint.com/tutorial/dip/images/gray-scale-images.png>

## 3. 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).



**Fig. Colour image**

<https://static.javatpoint.com/tutorial/dip/images/colour-images.png>

After understanding the basics of image and pixels, let us get started with exploring the concepts of Image processing.

## 5.5 Image Processing

### What is Image Processing?

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be an image or characteristics/features associated with that image.

### Why do we need it?

The data that we collect or generate is mostly raw data, i.e. it is not fit to be used in applications directly due to a number of possible reasons. Therefore, we need to analyze it first, perform the necessary pre-processing, and then use it.

For instance, let's assume that we were trying to build a cat classifier. Our program would take an image as input and then tell us whether the image contains a cat or not. The first step for building this classifier would be to collect hundreds of cat pictures. One common issue is that all the pictures we have scraped would not be of the same size/dimensions, so before feeding them to the model for training, we

would need to resize/pre-process them all to a standard size and apply filters on them to extract useful information.

This is just one of many reasons why image processing is essential to any computer vision

### **Image processing mainly include the following steps:**

- 1.Importing the image via image acquisition tools;
2. Analysing and manipulating the image;
- 3.Output in which result can be altered image or a report which is based on analysing that image.

## **Convolution**

Convolution is the process of adding each element of the image to its local neighbors, weighted by the kernel. This is related to a form of mathematical convolution. The matrix operation being performed—convolution—is not traditional matrix multiplication, despite being similarly denoted by \*.

For example, if we have two three-by-three matrices, the first a kernel, and the second an image piece, convolution is the process of flipping both the rows and columns of the kernel and multiplying locally similar entries and summing. The element at coordinates [2, 2] (that is, the central element) of the resulting image would be a weighted combination of all the entries of the image matrix, with weights given by the kernel:

$$\left( \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \right) [2, 2] = (i \cdot 1) + (h \cdot 2) + (g \cdot 3) + (f \cdot 4) + (e \cdot 5) + (d \cdot 6) + (c \cdot 7) + (b \cdot 8) + (a \cdot 9).$$

Fig : Convolution Formula

[https://wikimedia.org/api/rest\\_v1/media/math/render/svg/570600fdeed436d98626278f22bf034ff5ab5162](https://wikimedia.org/api/rest_v1/media/math/render/svg/570600fdeed436d98626278f22bf034ff5ab5162)

The other entries would be similarly weighted, where we position the center of the kernel on each of the boundary points of the image and compute a weighted sum.

## **Filters/Kernels in Image Processing**

In image processing, a kernel, convolution matrix, or mask is a small matrix used to apply effects like the ones you might find in Photoshop or Gimp. It is used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between a kernel and an image.

Depending on the element values, a kernel can cause a wide range of effects.



Fig : Unmodified (left) and the same image processed with a blur kernel.

<https://www.naturefocused.com/articles/kernel-blur-mask.jpg>

## Some popular types of kernels

**Edge detection:** this kernel detects edges within an image. A 3x3 example:

0	-1	0
-1	4	-1
0	-1	0

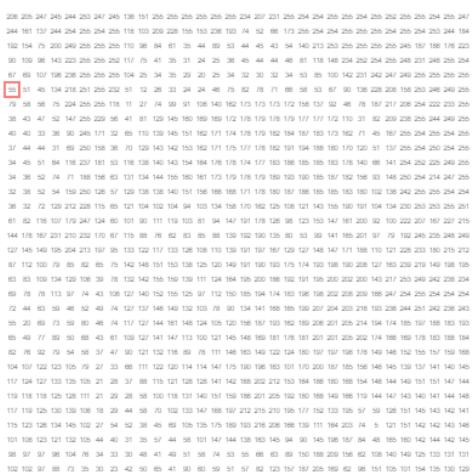
Fig : 3\*3 Filter

Notice that if all pixel values are comparable, the the resultant pixel value will be close to 0. However, edges - locations with extreme differences in pixel values - will result in values far from zero.

**Gaussian Blur:** This kernel is similar to the blur kernel presented above, but is different in that it is dependent upon the Gaussian function - a function which creates a distribution of values around the center point. This results in a kernel in which pixels near the center contribute more towards the new pixel value than those further away.

**Sharpening:** This kernel sharpens an image - accentuating the edges of the image. Sharpening an image adds contrast to edges, and a 3x3 version of this mask is similar to the edge detection kernel with a center value of 5. This adds contrast around an edge by accentuating bright and dark areas.

**Unsharp Mask:** Used to sharpen an image, this technique is based upon first creating a gaussian blurred copy of the image. This blurred copy is then subtracted from the original - pixels above a given threshold are sharpened by enhancing light and dark pixels.



Let's walk through applying the following 3x3 **sharpen** kernel to the image of a face from above.

**sharpen** ▾

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$



$$\begin{array}{r}
 ( \quad \boxed{255} + \boxed{232} + \boxed{5} \\
 \times 0 \qquad \times -1 \qquad \times 0 \\
 \\ 
 + \boxed{255} + \boxed{118} + \boxed{1} \\
 \times -1 \qquad \times 5 \qquad \times 0 \\
 \\ 
 + \boxed{229} + \boxed{56} + \boxed{41} \\
 \times 0 \qquad \times -1 \qquad \times 0 \\
 \\ 
 = \quad \boxed{36}
 \end{array}$$



Fig : Applying Kernel on Image

<https://setosa.io/ev/image-kernels/>

Check the above link for Image Kernels demo

## 5.6 Facial Detection

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. It is the action of locating human faces in an image and optionally returning different kinds of face-related data. At a minimum, each detected face corresponds to a faceRectangle field in the response. This set of pixel coordinates for the left, top, width, and height mark the located face. Using these coordinates, you can get the location of the face and its size.

Try the demo from this link

<https://azure.microsoft.com/en-us/services/cognitive-services/face/#demo>

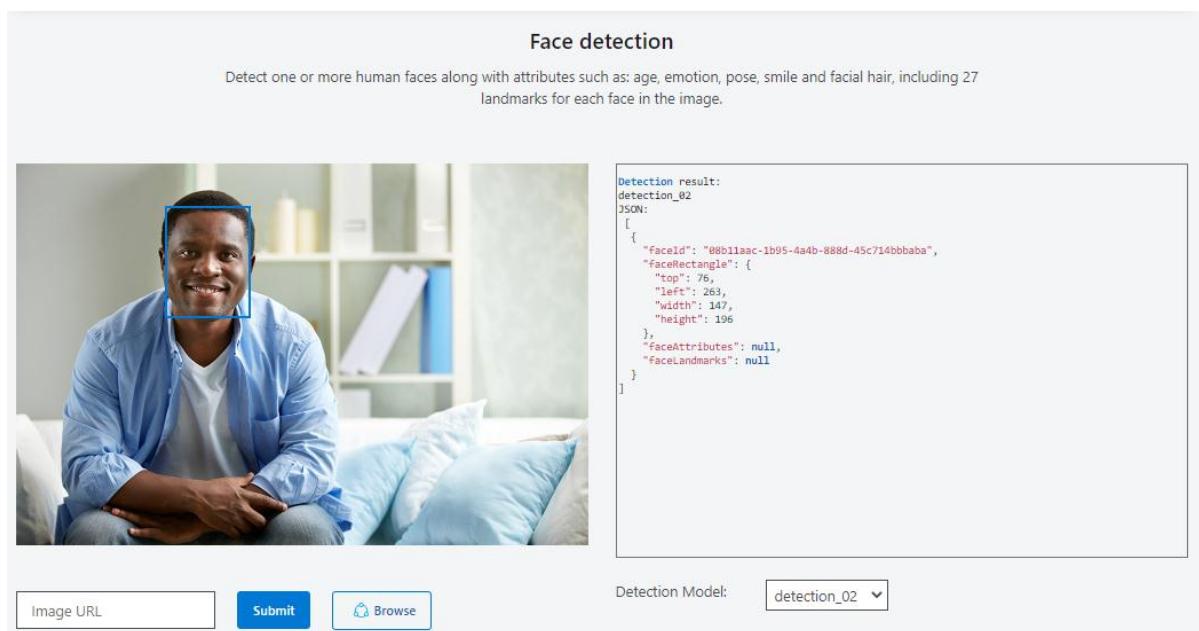
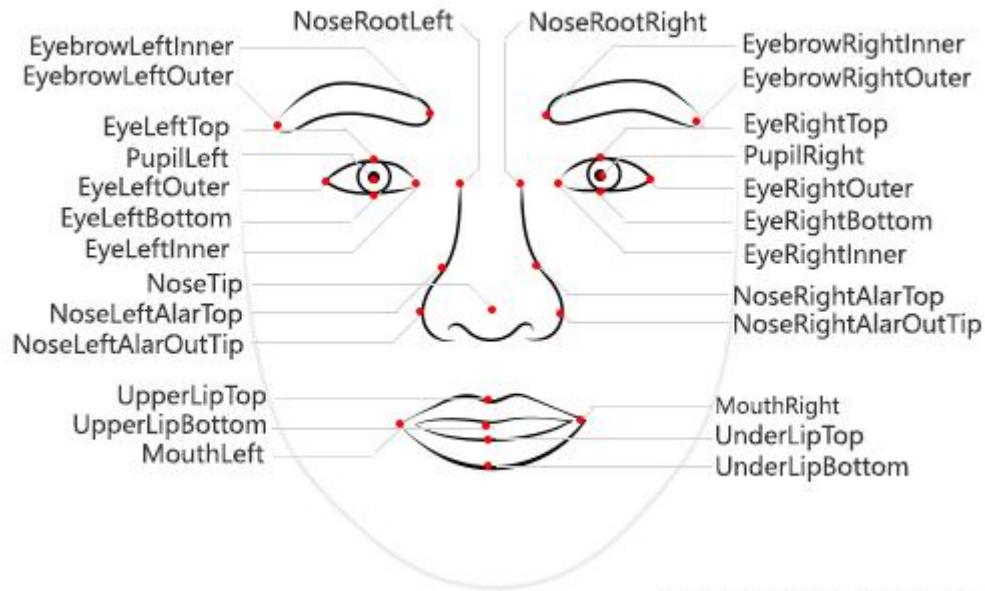


Fig : Face Detection Demo  
<https://azure.microsoft.com/en-us/services/cognitive-services/face/#demo>

## Face landmarks

Face landmarks are a set of easy-to-find points on a face, such as the pupils or the tip of the nose. By default, there are 27 predefined landmark points. The following figure shows all 27 points:



Copyright (c) Microsoft. All rights reserved

Fig : Face Landmarks

<https://docs.microsoft.com/en-us/azure/cognitive-services/face/images/landmarks.1.jpg>

## Face Detection With OpenCV

Feature-based face detection algorithms are fast and effective and have been used successfully for decades. Perhaps the most successful example is a technique called cascade classifiers first described by Paul Viola and Michael Jones and their 2001 paper titled “Rapid Object Detection using a Boosted Cascade of Simple Features.”

In the paper, effective features are learned using the AdaBoost algorithm, although importantly, multiple models are organized into a hierarchy or “cascade.”

Their detector, called detector cascade, consists of a sequence of simple-to-complex face classifiers and has attracted extensive research efforts. Moreover, detector cascade has been deployed in many commercial products such as smartphones and digital cameras.

A modern implementation of the Classifier Cascade face detection algorithm is provided in the OpenCV library. This is a C++ computer vision library that provides a python interface. The benefit of this implementation is that it provides pre-trained face detection models, and provides an interface to train a model on your own dataset.

OpenCV can be installed by the package manager system on your platform, or via pip; for example:

```
pip install opencv-python
```

OpenCV provides the `CascadeClassifier` class that can be used to create a cascade classifier for face detection. The constructor can take a filename as an argument that specifies the XML file for a pre-trained model.

OpenCV provides a number of pre-trained models as part of the installation. These are available on your system and are also available on the OpenCV GitHub project. Download a pre-trained model for frontal face detection from the OpenCV GitHub project and place it in your current working directory with the filename '`haarcascade_frontalface_default.xml`'.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

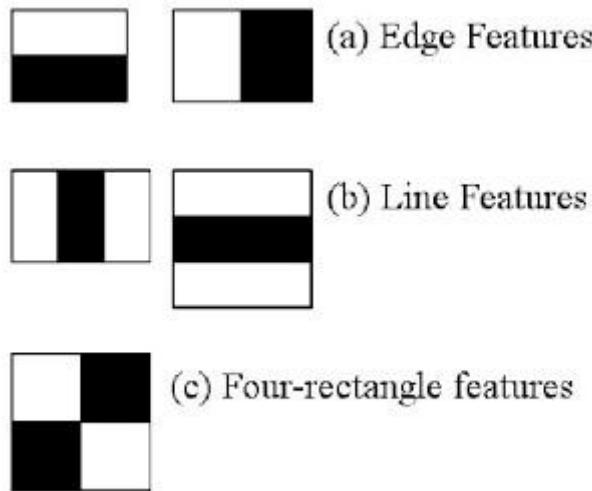


Fig : Features of an Image  
[https://docs.opencv.org/3.4.3/haar\\_features.jpg](https://docs.opencv.org/3.4.3/haar_features.jpg)

Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. Nice, isn't it? It makes things super-fast.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by **Adaboost**.

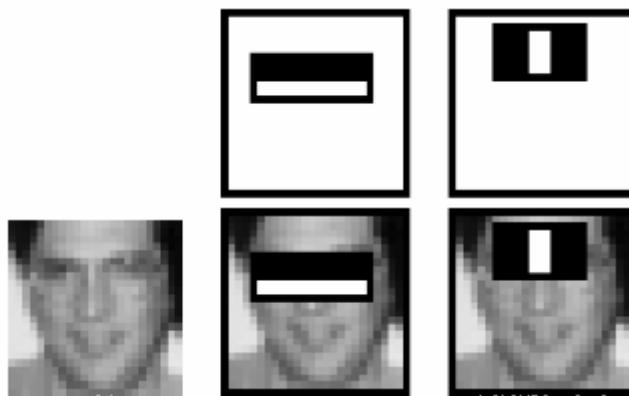


Fig : Applying Haarcascade on Faces  
<https://docs.opencv.org/3.4.3/haar.png>

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that most accurately classify the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then the same process is done. New error rates are calculated. Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features are found).

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).

So now you take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not. Wow.. Isn't it a little inefficient and time consuming? Yes, it is. The authors have a good solution for that.

In an image, most of the image is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a

single shot, and don't process it again. Instead, focus on regions where there can be a face. This way, we spend more time checking possible face regions.

For this they introduced the concept of **Cascade of Classifiers**. Instead of applying all 6000 features on a window, the features are grouped into different stages of classifiers and applied one-by-one. (Normally the first few stages will contain very many fewer features). If a window fails the first stage, discard it. We don't consider the remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

## Implementation of Face Detection in Code

```
faces = face_classifier.detectMultiScale(gray, 1.1, 6)
```

In this piece of code what we are trying to do is, using the **face\_classifier** which is an object loaded with **haarcascade\_frontalface\_default.xml**, we are using an inbuilt function with it called the **detectMultiScale**.

This function will help us to find the features/locations of the new image. The way it does is, it will use all the features from the **face\_classifier** object to detect the features of the new image.

The parameters that we will pass to this function are:

1. The gray scale variable — gray in our case
2. **scaleFactor** — Parameter specifying how much the image size is reduced at each image scale. Basically, the scale factor is used to create your scale pyramid. More explanation, your model has a fixed size defined during training, which is visible in the XML. This means that this size of the face is detected in the image if present. However, by rescaling the input image, you can resize a larger face to a smaller one, making it detectable by the algorithm. 1.05 is a good possible value for this, which means you use a small step for resizing, i.e. reduce the size by 5%, you increase the chance of a matching size with the model for detection is found.
3. **minNeighbors** — Parameter specifying how many neighbors each candidate rectangle should have to retain it. This parameter will affect the quality of the detected faces. Higher value results in fewer detections but with higher quality. 3~6 is a good value for it.

## Practical Code for CV Basics and Image Processing:

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/10.%20OpenCV%20Basics.ipynb>

After understanding the basics of image processing let us start with a multi-layered network convolutional neural network with special architecture to detect the feature from data and which is used in computer vision applications and self-driving cars.

## 5.7 Convolutional Neural Networks

### What are CNNs?

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery

A convolutional neural network is a particularly effective artificial neural network, and it presents a unique architecture. Layers are organized in three dimensions: width, height, and depth. The neurons in one layer connect not to all the neurons in the next layer, but only to a small region of the layer's neurons. The final output is reduced to a single vector of probability scores, organized along the depth dimension.

Convolutional neural networks have been used in areas such as video recognition, image recognition, and recommender systems.

Convolutional networks were inspired by biological processes<sup>[9][10][11][12]</sup> in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns to optimize the filters (or kernels) through automated learning, whereas in traditional algorithms these filters are hand-engineered. This independence from prior knowledge and human intervention in feature extraction is a major advantage.

The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution which is discussed in the beginning of chapter. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

## Building Blocks of Convolutional Neural Networks

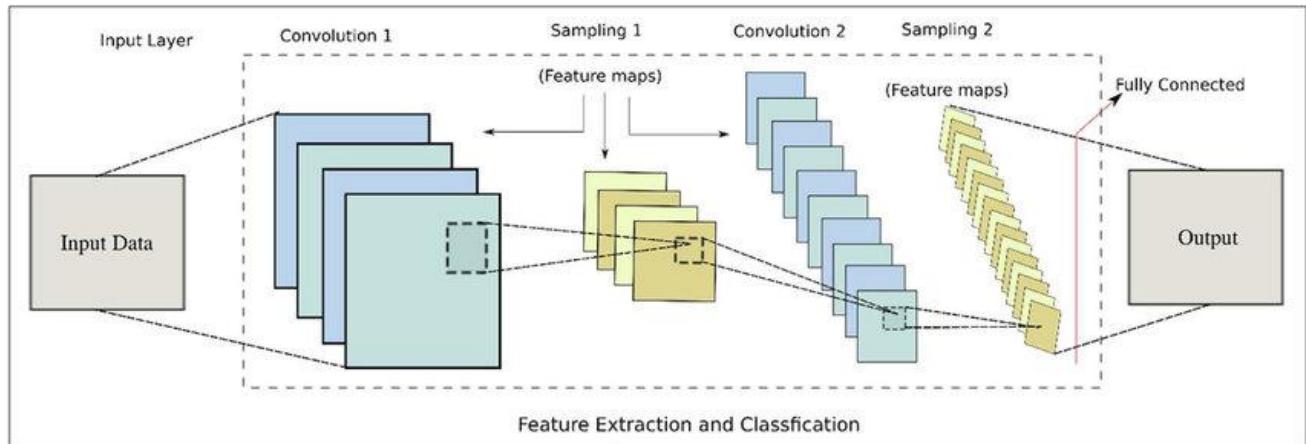


Fig : Architecture of a Convolutional Neural Network

[https://www.researchgate.net/profile/Haleem\\_Farman/publication/321787151/figure/fig3/AS:673901872951307@1537682135746/CNN-general-architecture.jpg](https://www.researchgate.net/profile/Haleem_Farman/publication/321787151/figure/fig3/AS:673901872951307@1537682135746/CNN-general-architecture.jpg)

There are three types of layers in a Convolutional Neural Network:

### 1. Convolutional layers

Convolutional layers are the major building blocks used in convolutional neural networks.

A convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image.

The innovation of convolutional neural networks is the ability to automatically learn a large number of filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification. The result is highly specific features that can be detected anywhere on input images.

The output from multiplying the filter with the input array one time is a single value. As the filter is applied multiple times to the input array, the result is a two-dimensional array of output values that represent a filtering of the input. As such, the two-dimensional output array from this operation is called a “*feature map*”.

Once a feature map is created, we can pass each value in the feature map through a nonlinearity, such as a ReLU, much like we do for the outputs of a fully connected layer.

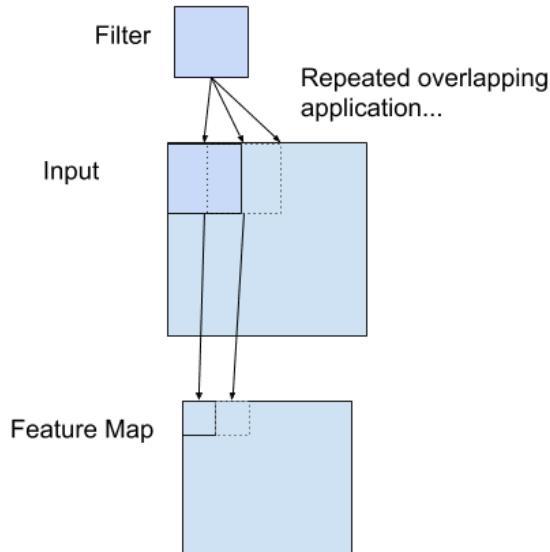


Fig : Example of a Filter Applied to a Two-Dimensional Input to Create a Feature Map  
<https://machinelearningmastery.com/wp-content/uploads/2019/01/Example-of-a-Filter-Applied-to-a-Two-Dimensional-input-to-create-a-Feature-Map.png>

### Convolutional Layers in Keras

» [Keras API reference / Layers API / Convolution layers](#)

---

## Convolution layers

- [Conv1D layer](#)
- [Conv2D layer](#)
- [Conv3D layer](#)
- [SeparableConv1D layer](#)
- [SeparableConv2D layer](#)
- [DepthwiseConv2D layer](#)
- [Conv2DTranspose layer](#)
- [Conv3DTranspose layer](#)

Fig : Convolutional Layers in Keras  
[https://keras.io/api/layers/convolution\\_layers/](https://keras.io/api/layers/convolution_layers/)

The layer used for convolution of images is 2D Convolution layer. Most important parameters of Conv2D Layer are:

### ***Filters***

The first required Conv2D parameter is the number of filters that the convolutional layer will learn. Layers early in the network architecture (i.e., closer to the actual input image) learn fewer convolutional filters while layers deeper in the network (i.e., closer to the output predictions) will learn more filters. Conv2D layers in between will learn more filters than the early Conv2D layers but fewer filters than the layers closer to the output. Max pooling is then used to reduce the spatial dimensions of the output volume.

### ***Kernel Size***

The second required parameter you need to provide to the Keras Conv2D class is the `kernel_size` , a 2-tuple specifying the width and height of the 2D convolution window. The `kernel_size` must be an odd integer as well. Typical values for `kernel_size` include: `(1, 1)` , `(3, 3)` , `(5, 5)` , `(7, 7)` . It's rare to see kernel sizes larger than `7x7`.

### ***Strides***

The strides parameter is a 2-tuple of integers, specifying the “step” of the convolution along the x and y axis of the input volume.

The strides value defaults to `(1, 1)` , implying that:

1. A given convolutional filter is applied to the current location of the input volume
2. The filter takes a 1-pixel step to the right and again the filter is applied to the input volume
3. This process is performed until we reach the far-right border of the volume in which we move our filter one pixel down and then start again from the far left.

Typically you'll leave the strides parameter with the default `(1, 1)` value; however, you may occasionally increase it to `(2, 2)` to help reduce the size of the output volume (since the step size of the filter is larger).

### ***Padding***

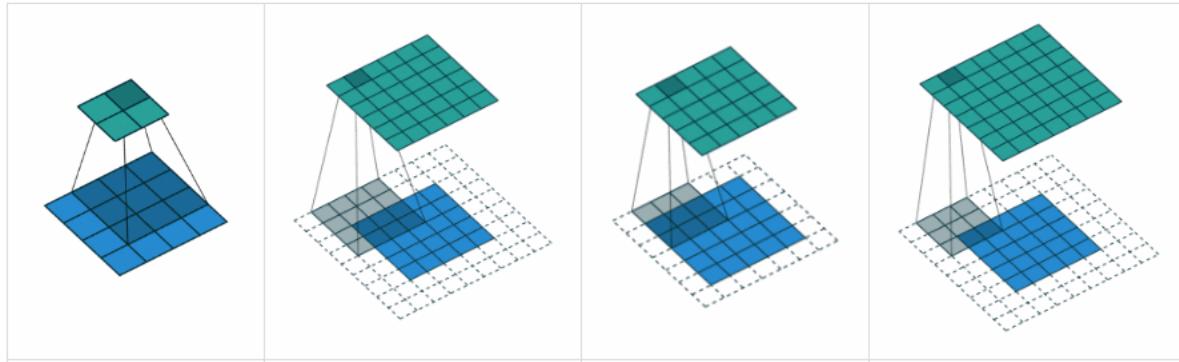


Fig : Padding in CNN  
<https://i.stack.imgur.com/0rs9I.gif>

If the size of the previous layer is not cleanly divisible by the size of the filters receptive field and the size of the stride then it is possible for the receptive field to attempt to read off the edge of the input feature map. In this case, techniques like zero padding can be used to invent mock inputs for the receptive field to read.

The padding parameter to the Keras Conv2D class can take on one of two values: valid or same .

- Padding '**valid**' is the first figure. The filter window stays inside the image.
- Padding '**same**' is the third figure. The output is the same size.

## 2. Pooling Layers

The pooling layers down-sample the previous layers feature map.

Pooling layers follow a sequence of one or more convolutional layers and are intended to consolidate the features learned and expressed in the previous layers feature map.

As such, pooling may be considering a technique to compress or generalize feature representations and generally reduce the overfitting of the training data by the model.

They too have a receptive field, often much smaller than the convolutional layer. Also, the stride or number of inputs that the receptive field is moved for each activation is often equal to the size of the receptive field to avoid any overlap.

Pooling layers are often very simple, taking the average or the maximum of the input value in order to create its own feature map.

The pooling operation is specified, rather than learned. Two common functions used in the pooling operation are:

- **Average Pooling:** Calculate the average value for each patch on the feature map.
- **Maximum Pooling (or Max Pooling):** Calculate the maximum value for each patch of the feature map.

The result of using a pooling layer and creating down sampled or pooled feature maps is a summarized version of the features detected in the input. They are useful as small changes in the location of the feature in the input detected by the convolutional layer will result in a pooled feature map with the feature in the same location.

### *Max Pooling Layer*

Maximum pooling, or max pooling, is a pooling operation that calculates the maximum, or largest, value in each patch of each feature map.

The results are down sampled or pooled feature maps that highlight the most present feature in the patch, not the average presence of the feature in the case of average pooling. This has been found to work better in practice than average pooling for computer vision tasks like image classification.

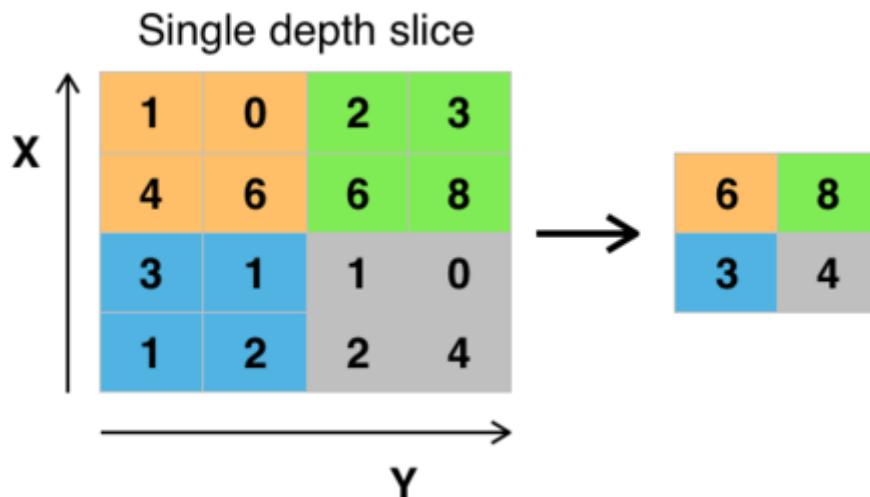


Fig : Max Pooling Layer in CNN

[https://upload.wikimedia.org/wikipedia/commons/thumb/e/e9/Max\\_pooling.png/471px-Max\\_pooling.png](https://upload.wikimedia.org/wikipedia/commons/thumb/e/e9/Max_pooling.png/471px-Max_pooling.png)

In modern CNNs, *max pooling* is typically used, and often of size 2x2, with a stride of 2. This implies that the input is drastically downsampled, further improving the computational efficiency.

## 3. Fully Connected Layers

Fully connected layers are the normal flat feed-forward neural network layer.

These layers may have a non-linear activation function or a softmax activation in order to output probabilities of class predictions.

Fully connected layers are used at the end of the network after feature extraction and consolidation has been performed by the convolutional and pooling layers. They are used to create final non-linear combinations of features and for making predictions by the network.

Now we got introduced with artificial neural networks and convolutional neural networks, let's get started with another deep learning technique named transfer learning.

### Practical code for CNN:

This project demonstrates training a simple Convolutional Neural Network (CNN) to classify CIFAR images. Because this tutorial uses the Keras Sequential API, creating and training your model will take just a few lines of code. The CIFAR10 dataset contains 60,000 color images in 10 classes, with 6,000 images in each class. The dataset is divided into 50,000 training images and 10,000 testing images. The classes are mutually exclusive and there is no overlap between them.

Link for code:

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/11.%20CNN%20-%20CIFAR%2010.ipynb>

## 5.8 What is transfer learning?

Training deep learning models often requires large amounts of training data, high-end compute resources (GPU, TPU), and a longer training time. In scenarios when you don't have any of these available to you, you can shortcut the training process using a technique known as *transfer learning*.

Transfer learning is a technique that applies knowledge gained from solving one problem to a different but related problem.

Due to the structure of neural networks, the first set of layers usually contain lower-level features, whereas the final set of layers contains higher-level feature that are closer to the domain in question. By repurposing the final layers for use in a new domain or problem, you can significantly reduce the amount of time, data, and compute resources needed to train the new model. For example, if you already have

a model that recognizes cars, you can repurpose that model using transfer learning to also recognize trucks, motorcycles, and other kinds of vehicles.

## History of Transfer Learning

In 1976 Stevo Bozinovski and Ante Fulgosi published a paper explicitly addressing transfer learning in neural networks training. The paper gives a mathematical and geometrical model of transfer learning. In 1981 a report was given on the application of transfer learning in training a neural network on a dataset of images representing letters of computer terminals. Both positive and negative transfer learning was experimentally demonstrated

In 1993, Lorien Pratt published a paper on transfer in machine learning, formulating the discriminability-based transfer (DBT) algorithm

In 1997, Pratt and Sebastian Thrun guest edited a special issue of Machine Learning devoted to transfer learning, and by 1998, the field had advanced to include multi-task learning, along with a more formal analysis of its theoretical foundations. Learning to Learn, edited by Thrun and Pratt, is a 1998 review of the subject.

Transfer learning has also been applied in cognitive science, with Pratt also guest editing an issue of Connection Science on reuse of neural networks through transfer in 1996.

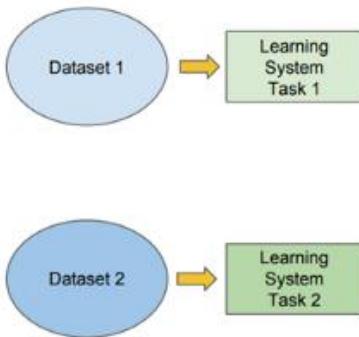
Andrew Ng said in his NIPS 2016 tutorial that TL will be the next driver of ML commercial success after supervised learning to highlight the importance of TL.

## Understanding Transfer Learning

The first thing to remember here is that, transfer learning, is not a new concept which is very specific to deep learning. There is a stark difference between the traditional approach of building and training machine learning models, and using a methodology following transfer learning principles.

## Traditional ML vs Transfer Learning

- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks



- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data

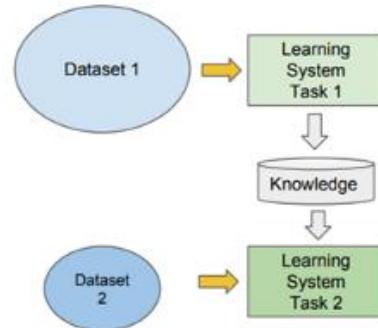


Fig : Traditional ML vs Transfer Learning  
[https://miro.medium.com/max/700/1\\*9GTEzcO8KxxrfutmtsPs3Q.png](https://miro.medium.com/max/700/1*9GTEzcO8KxxrfutmtsPs3Q.png)

Traditional learning is isolated and occurs purely based on specific tasks, datasets and training separate isolated models on them. No knowledge is retained which can be transferred from one model to another. In transfer learning, you can leverage knowledge (features, weights etc) from previously trained models for training newer models and even tackle problems like having less data for the newer task!

Let's understand the preceding explanation with the help of an example. Let's assume our task is to identify objects in images within a restricted domain of a restaurant. Let's mark this task in its defined scope as **T1**. Given the dataset for this task, we train a model and tune it to perform well (generalize) on unseen data points from the same domain (restaurant). Traditional supervised ML algorithms break down when we do not have sufficient training examples for the required tasks in given domains. Suppose, we now must detect objects from images in a park or a café (say, task **T2**). Ideally, we should be able to apply the model trained for **T1**, but in reality, we face performance degradation and models that do not generalize well. This happens for a variety of reasons, which we can liberally and collectively term as the model's bias towards training data and domain.

Transfer learning should enable us to utilize knowledge from previously learned tasks and apply them to newer, related ones. If we have significantly more data for task **T1**, we may utilize its learning, and generalize this knowledge (features, weights) for task **T2** (which has significantly less data). In the case of problems in the computer

vision domain, certain low-level features, such as edges, shapes, corners and intensity, can be shared across tasks, and thus enable knowledge transfer among tasks! Also, as we have depicted in the earlier figure, knowledge from an existing task acts as an additional input when learning a new target task.

## Transfer learning: idea

Instead of training a deep network from scratch for your task:

- Take a network trained on a different domain for a different **source task**
- Adapt it for your domain and your **target task**

Variations:

- Same domain, different task
- Different domain, same task

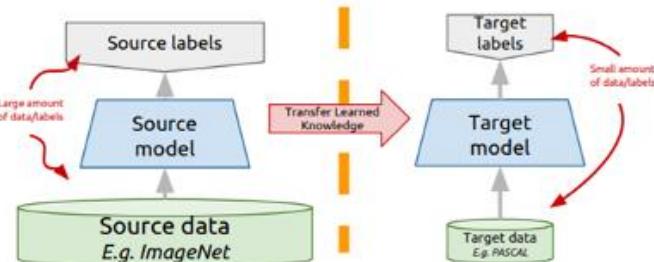


Fig : Transfer Learning Idea

[https://miro.medium.com/max/700/1\\*Ww3AMxZeoiB84GVSRRBr4Bw.png](https://miro.medium.com/max/700/1*Ww3AMxZeoiB84GVSRRBr4Bw.png)

## Pre-trained Models

One of the fundamental requirements for transfer learning is the presence of models that perform well on source tasks. Luckily, the deep learning world believes in sharing. Many of the state-of-the art deep learning architectures have been openly shared by their respective teams. These span across different domains, such as computer vision and NLP, the two most popular domains for deep learning applications. Pre-trained models are usually shared in the form of the millions of parameters/weights the model achieved while being trained to a stable state. Pre-trained models are available for everyone to use through different means. The famous deep learning Python library, keras, provides an interface to download some popular models. You can also access pre-trained models from the web since most of them have been open-sourced.

## Transfer Learning with Image Data

It is common to perform transfer learning with predictive modeling problems that use image data as input.

This may be a prediction task that takes photographs or video data as input.

For these types of problems, it is common to use a deep learning model pre-trained for a large and challenging image classification task such as the ImageNet 1000-class photograph classification competition.

The [ImageNet Large Scale Visual Recognition Challenge](#) or ILSVRC for short is an annual competition held between 2010 and 2017 in which challenge tasks use subsets of the ImageNet dataset.

The goal of the challenge was to both promote the development of better computer vision techniques and to benchmark the state of the art.

The annual challenge focussed on multiple tasks for “*image classification*” that includes both assigning a class label to an image based on the main object in the photograph and “*object detection*” that involves localizing objects within the photograph.



Fig: ImageNet Competition

[https://miro.medium.com/max/750/1\\*IIZW43-NtJrwqtt5Xy3ISA.jpeg](https://miro.medium.com/max/750/1*IIZW43-NtJrwqtt5Xy3ISA.jpeg)

The research organizations that develop models for this competition and do well often release their final model under a permissive license for reuse. These models can take days or weeks to train on modern hardware.

These models can be downloaded and incorporated directly into new models that expect image data as input.

**For computer vision**, you can leverage some popular models which are included in Keras deep learning library,

- VGG-16
- Inception V3
- ResNet-50

### *Inception (GoogLeNet)*

Christian Szegedy, et al. from Google achieved top results for object detection with their GoogLeNet model that made use of the inception module and architecture. This approach was described in their 2014 paper titled “Going Deeper with Convolutions.”

*We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14).*

— Going Deeper with Convolutions, 2014.

### VGG

Karen Simonyan and Andrew Zisserman from the Oxford Vision Geometry Group (VGG) achieved top results for image classification and localization with their VGG model. Their approach is described in their 2015 paper titled “Very Deep Convolutional Networks for Large-Scale Image Recognition.”

*... we come up with significantly more accurate ConvNet architectures, which not only achieve the state-of-the-art accuracy on ILSVRC classification and localisation tasks, but are also applicable to other image recognition datasets, where they achieve excellent performance even when used as a part of a relatively simple pipelines*

— Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015.

### *ResNet (MSRA)*

Kaiming He, et al. from Microsoft Research achieved top results for object detection and object detection with localization tasks with their Residual Network or ResNet described in their 2015 paper titled “Deep Residual Learning for Image Recognition.”

*An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task.*

— Deep Residual Learning for Image Recognition, 2015.

These models are both widely used for transfer learning both because of their performance, but also because they were examples that introduced specific architectural innovations, namely consistent and repeating structures (VGG), inception modules (GoogLeNet), and residual modules (ResNet).

Keras provides access to a number of top-performing pre-trained models that were developed for image recognition tasks.

They are available via the Applications API, and include functions to load a model with or without the pre-trained weights, and prepare data in a way that a given model may expect (e.g. scaling of size and pixel values).

The first time a pre-trained model is loaded, Keras will download the required model weights, which may take some time given the speed of your internet connection. Weights are stored in the `.keras/models/` directory under your home directory and will be loaded from this location the next time that they are used.

When loading a given model, the “`include_top`” argument can be set to `False`, in which case the fully-connected output layers of the model used to make predictions is not loaded, allowing a new output layer to be added and trained.

Now we understood the concepts of transfer learning which saves a lot of time in terms of training networks, let us go to our next concept of optical character recognition.

## Practical code for Transfer Learning:

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/12.%20Transfer%20Learning%20-%20MobileNet.ipynb>

## 5.9 Optical Character Recognition

OCR systems transform a two-dimensional image of text, that could contain machine printed or handwritten text from its image representation into machine-readable text. OCR as a process generally consists of several sub-processes to perform as accurately as possible. The subprocesses are:

- Preprocessing of the Image
- Text Localization
- Character Segmentation

- Character Recognition
- Post Processing

The sub-processes in the list above of course can differ, but these are roughly the steps needed to approach automatic character recognition. In OCR software, it's main aim is to identify and capture all the unique words using different languages from written text characters.

For almost two decades, optical character recognition systems have been widely used to provide automated text entry into computerized systems. Yet in all this time, conventional OCR systems have never overcome their inability to read more than a handful of type fonts and page formats. Proportionally spaced type (which includes virtually all typeset copy), laser printer fonts, and even many non-proportional typewriter fonts, have remained beyond the reach of these systems. And as a result, conventional OCR has never achieved more than a marginal impact on the total number of documents needing conversion into digital form.



Fig : Working of OCR  
<https://nanonets.com/blog/content/images/2019/11/OCR.jpg>

Next-generation OCR engines deal with these problems mentioned above really good by utilizing the latest research in the area of deep learning. By leveraging the combination of deep models and huge datasets publicly available, models achieve state-of-the-art accuracies on given tasks. Nowadays it is also possible to generate synthetic data with different fonts using generative adversarial networks and few other generative approaches.

Optical Character Recognition remains a challenging problem when text occurs in unconstrained environments, like natural scenes, due to geometrical distortions,

complex backgrounds, and diverse fonts. The technology still holds an immense potential due to the various use-cases of deep learning based OCR like

- building license plate readers
- digitizing invoices
- digitizing menus
- digitizing ID cards

Lets see the technology behind the most used Tesseract Engine, which was upgraded with the latest knowledge researched in optical character recognition. This article will also serve as a how-to guide/ tutorial on how to implement OCR in python using the Tesseract engine.

Tesseract - an open-source OCR engine that has gained popularity among OCR developers. Even though it can be painful to implement and modify sometimes, there weren't too many free and powerful OCR alternatives on the market for the longest time. Tesseract began as a Ph.D. research project in HP Labs, Bristol. It gained popularity and was developed by HP between 1984 and 1994. In 2005 HP released Tesseract as an open-source software.

## **Tesseract OCR**

Tesseract is an open source text recognition (OCR) Engine, available under the Apache 2.0 license. It can be used directly, or (for programmers) using an API to extract printed text from images. It supports a wide variety of languages. Tesseract doesn't have a built-in GUI, but there are several available from the 3<sup>rd</sup> Party page. Tesseract is compatible with many programming languages and frameworks through wrappers that can be found here. It can be used with the existing layout analysis to recognize text within a large document, or it can be used in conjunction with an external text detector to recognize text from an image of a single text line.

## OCR Process Flow

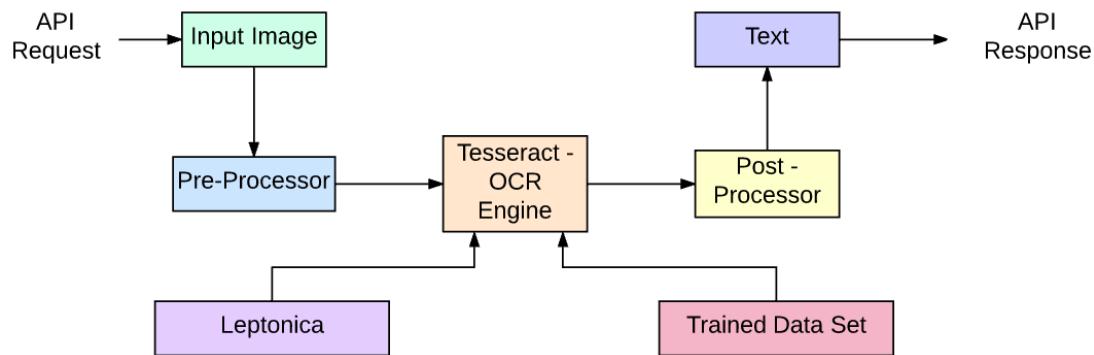


Fig : OCR Process Flow

## OCR with Pytesseract and OpenCV

Pytesseract is a wrapper for Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others.

Install the library using pip install pytesseract

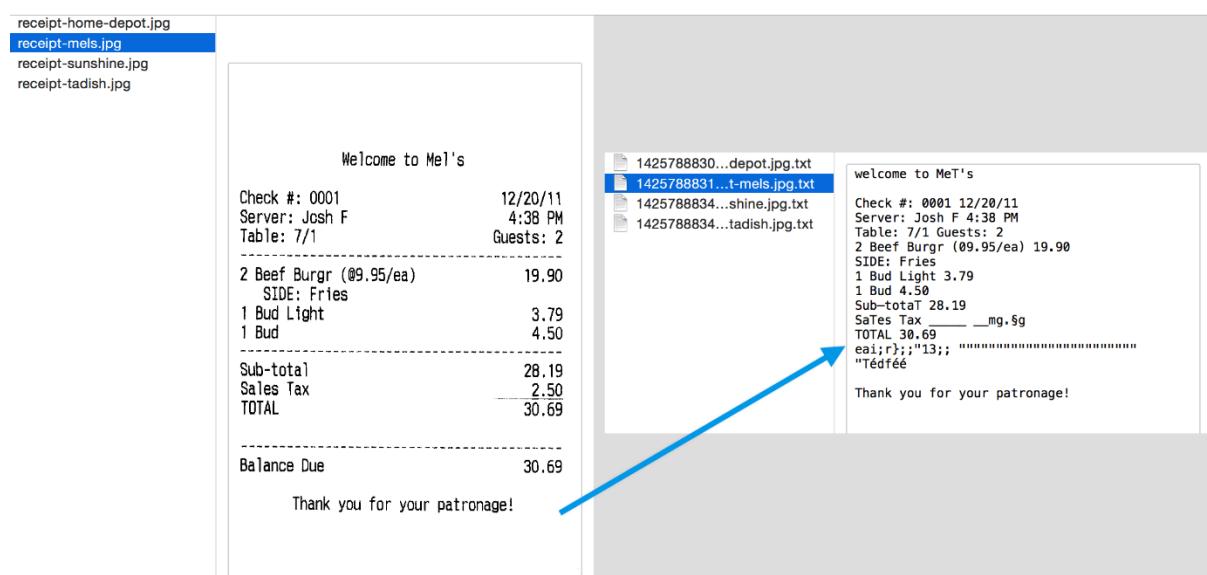


Fig : Demonstration of OCR

[https://nanonets.com/blog/content/images/2019/11/ocr\\_img.png](https://nanonets.com/blog/content/images/2019/11/ocr_img.png)

## Practical Code for OCR

To accomplish OCR with Python on Windows, you will need Python and OpenCV which you already have, as well as Tesseract and the Pytesseract Python package.

To install Tesseract OCR for Windows:

1. Run the installer from UB Mannheim

Link : <https://github.com/UB-Mannheim/tesseract/wiki>

2. Make sure that this exe is installed in C:\Program Files (x86)\Tesseract- OCR

3. You will need to add the following line in your code in order to be able to call pytesseract on your machine: `pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'`

We will see how to apply OCR on the image which contains quote. We will checkss results in 2 variants.

1. Specific characters along with Bounding boxes
2. Specific words with Bounding boxes



Link for the project

<https://github.com/Edunet-Foundation/Tech-Saksham/blob/main/Tech%20Saksham%20AI%20Practical%20-%20Notebooks/Chapter%205/13.%20OCR%20-%20Tesseract.ipynb>

So, in this chapter we explored the concepts of deep learning like artificial neural networks, image processing, convolutional neural networks, transfer learning and optical character recognition. Happy Learning.

Please refers below mentioned Links to Codes/ Practical Notebooks of all the chapters

1. [Chapter 2](#)
2. [Chapter 3](#)
3. [Chapter 4](#)
4. [Chapter 5](#)

## Reference

1. [Public information, Deloitte Research](#)
2. <http://www.oreilly.com/data/free/the-new-artificial-intelligence-market.csp>
3. <https://www.weforum.org/agenda/2018/09/artificial-intelligence-shaking-up-job-market/>
4. [https://en.wikiversity.org/wiki/Artificial\\_intelligence/Introduction](https://en.wikiversity.org/wiki/Artificial_intelligence/Introduction)
5. <https://techvidvan.com/tutorials/artificial-intelligence-applications/>
6. <https://www.xenonstack.com/blog/machine-learning-pipeline/>
7. <https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/ai-overview>
8. <https://www.javatpoint.com/history-of-artificial-intelligence>
9. <https://aidemos.microsoft.com/>
10. <https://lobe.ai/>
11. [https://www.tutorialspoint.com/operating\\_system/os\\_linux.html](https://www.tutorialspoint.com/operating_system/os_linux.html)
12. <https://buildmedia.readthedocs.org/media/pdf/lym/latest/lym.pdf>
13. <https://phoenixnap.com/kb/linux-commands-cheat-sheet>
14. <https://www.guru99.com/file-permissions.html>
15. <https://www.hostinger.in/tutorials/linux-commands>
16. <https://www.guru99.com/introduction-to-shell-scripting.html>
17. <https://www.geeksforgeeks.org/cat-command-in-linux-with-examples/>
18. <https://www.geeksforgeeks.org/basic-shell-commands-in-linux/>
19. <https://www.tecmint.com/13-basic-cat-command-examples-in-linux/>
20. <https://phoenixnap.com/kb/use-nano-text-editor-commands-linux>
21. <https://linuxize.com/post/how-to-use-nano-text-editor/>
22. <https://www.javatpoint.com/how-to-install-vi-editor-in-ubuntu>
23. <http://www.compciv.org/recipes/cli/basic-shell-scripts/>
24. [https://en.wikipedia.org/wiki/Anaconda\\_\(Python\\_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution))
25. <https://www.python.org/doc/essays/blurb/>

26. [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
27. <https://docs.python.org/3/library/>
28. <https://www.python.org/downloads/release/python-394/>
29. <https://www.datacamp.com/community/tutorials/functions-python-tutorial>
30. <https://www.tutorialspoint.com/difference-between-method-and-function-in-python>
31. [https://www.w3schools.com/python/python\\_conditions.asp](https://www.w3schools.com/python/python_conditions.asp)
32. <https://www.geeksforgeeks.org/python-data-types/>
33. [https://www.tutorialspoint.com/python/python\\_strings.htm](https://www.tutorialspoint.com/python/python_strings.htm)
34. <https://www.datacamp.com/community/tutorials/python-string-format>
35. <https://docs.python.org/3/library/>
36. <https://www.tutorialspoint.com/What-are-Python-function-attributes>
37. <https://towardsdatascience.com/>
38. [https://en.wikipedia.org/wiki/Data\\_analysis](https://en.wikipedia.org/wiki/Data_analysis)
39. <https://techdifferences.com/difference-between-descriptive-and-predictive-data-mining.html>
40. <https://www.tutorialspoint.com/numpy>
41. <https://numpy.org/>
42. [https://www.tutorialspoint.com/python\\_data\\_science](https://www.tutorialspoint.com/python_data_science)
43. <https://www.geeksforgeeks.org/generating-random-number-list-in-python/>
44. [https://www.w3schools.com/python\(numpy\\_random.asp](https://www.w3schools.com/python(numpy_random.asp)
45. <https://www.geeksforgeeks.org/numpy-asscalar-in-python/>
46. <https://data-flair.training/blogs/numpy-statistical-functions/>
47. <https://www.w3schools.in/python-tutorial/decision-making/>
48. <https://www.geeksforgeeks.org/python-broadcasting-with-numpy-arrays/>
49. <https://realpython.com/>
50. <https://www.tableau.com/learn/articles/data-visualization#:~:text=Data%20visualization%20is%20the%20graphical,outliers%2C%20and%20patterns%20in%20data>
51. <https://towardsdatascience.com/introduction-to-data-visualization-in-python-89a54c97fbed>
52. [https://www.w3schools.com/python/matplotlib\\_subplots.asp](https://www.w3schools.com/python/matplotlib_subplots.asp)
53. [https://www.w3schools.com/python/matplotlib\\_markers.asp](https://www.w3schools.com/python/matplotlib_markers.asp)
54. <https://matplotlib.org/stable/users/installing.html>
55. <https://www.geeksforgeeks.org/graph-plotting-in-python-set-1/>
56. <https://www.geeksforgeeks.org/python-matplotlib-pyplot-ticks/>
57. <https://www.geeksforgeeks.org/matplotlib-pyplot-legend-in-python/>
58. <https://jakevdp.github.io/PythonDataScienceHandbook/04.06-customizing-legends.html>
59. <https://www.oreilly.com/library/view/python-data-science/9781491912126/ch04.html>
60. <https://jakevdp.github.io/PythonDataScienceHandbook/04.05-histograms-and-binnings.html>

61. <https://jakevdp.github.io/PythonDataScienceHandbook/04.09-text-and-annotation.html>
62. <https://jakevdp.github.io/PythonDataScienceHandbook/04.12-three-dimensional-plotting.html>
63. <https://www.geeksforgeeks.org/three-dimensional-plotting-in-python-using-matplotlib/>
64. <https://pythontic.com/visualization/charts/piechart>
65. <https://www.geeksforgeeks.org/plot-a-pie-chart-in-python-using-matplotlib/>
66. [https://www.tutorialspoint.com/numpy/numpy\\_matplotlib.htm](https://www.tutorialspoint.com/numpy/numpy_matplotlib.htm):
67. <https://www.edureka.co/blog/python-numpy-tutorial/>
68. [https://pandas.pydata.org/docs/getting\\_started/index.html](https://pandas.pydata.org/docs/getting_started/index.html)
69. <https://datatofish.com/install-package-python-using-pip/>
70. <https://packaging.python.org/>
71. <https://jakevdp.github.io/PythonDataScienceHandbook/03.01-introducing-pandas-objects.html>
72. [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/io.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html)
73. <https://www.bigcommerce.com/ecommerce-answers/what-csv-file-and-what-does-it-mean-my-ecommerce-business/#:~:text=A%20CSV%20is%20a%20comma,Microsoft%20Excel%20or%20Google%20Spreadsheets.>
74. <https://fileinfo.com/extension/json>
75. <https://www.w3resource.com/JSON/structures.php>
76. <https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files/>
77. <https://datatofish.com/load-json-pandas-dataframe/>
78. <https://www.shanelynn.ie/summarising-aggregation-and-grouping-data-in-python-pandas>
79. <https://towardsdatascience.com/pandas-groupby-explained-453692519d0>
80. <https://towardsdatascience.com/a-step-by-step-guide-to-pandas-pivot-tables-e0641d0c6c70>
81. [https://pandas.pydata.org/docs/reference/api/pandas.pivot\\_table.html](https://pandas.pydata.org/docs/reference/api/pandas.pivot_table.html)
82. <https://datatofish.com/plot-dataframe-pandas>
83. <https://towardsdatascience.com/an-introduction-to-scikit-learn-the-gold-standard-of-python-machine-learning-e2b9238a98ab>
84. <https://www.javatpoint.com/regression-vs-classification-in-machine-learning>
85. <https://towardsdatascience.com/mathematics-for-machine-learning-linear-regression-least-square-regression-de09cf53757c>
86. <https://www.javatpoint.com/linear-regression-in-machine-learning>
87. <https://www.i2tutorials.com/tag/ordinary-least-square-method-in-machine-learning/>
88. <https://www.javatpoint.com/logistic-regression-in-machine-learning>
89. <https://www.datasciencecentral.com/profiles/blogs/understanding-the-applications-of-probability-in-machine-learning>
90. <https://www.allerin.com/blog/how-to-fine-tune-your-artificial-intelligence-algorithms>

91. <https://www.mygreatlearning.com/blog/gridsearchcv/>
92. <https://www.kdnuggets.com/2020/05/hyperparameter-optimization-machine-learning-models.html>
93. <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>
94. <https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589>
95. <https://medium.com/analytics-vidhya/role-of-distance-metrics-in-machine-learning-e43391a6bf2e>
96. <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
97. <https://towardsdatascience.com/basic-probability-theory-and-statistics-3105ab637213>
98. <https://www.javatpoint.com/machine-learning-naive-bayes-classifier>
99. Andreas C. Müller and Sarah Guido , Introduction to Machine learning with Python , O'reilly , October 2016.
100. <https://www.guru99.com/unsupervised-machine-learning.html#:~:text=Unsupervised%20Learning%20is%20a%20machine,deals%20with%20the%20unlabelled%20data.>
101. <https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a>
102. <https://www.geeksforgeeks.org/clustering-in-machine-learning/#:~:text=Clustering%20is%20the%20task%20of,data%20points%20in%20other%20groups.>
103. <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>
104. Chire, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0>>, via Wikimedia Commons\
105. [https://en.wikipedia.org/wiki/Deep\\_learning](https://en.wikipedia.org/wiki/Deep_learning)
106. <https://www.geeksforgeeks.org/difference-between-machine-learning-and-deep-learning/>
107. [https://en.wikipedia.org/wiki/Artificial\\_neural\\_network](https://en.wikipedia.org/wiki/Artificial_neural_network)
108. <https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f>
109. <https://ml-cheatsheet.readthedocs.io/en/latest/forwardpropagation.html>
110. <https://en.wikipedia.org/wiki/Overfitting>
111. <https://medium.com/analytics-vidhya/the-perfect-fit-for-a-dnn-596954c9ea39>
112. <https://www.kdnuggets.com/2017/11/create-good-validation-set.html>
113. <https://medium.datadriveninvestor.com/data-science-essentials-why-train-validation-test-data-b7f7d472dc1f>
114. <https://machinelearningmastery.com/what-is-computer-vision/>
115. <https://en.wikipedia.org/wiki/Pixel>
116. <https://www.javatpoint.com/dip-types-of-images>
117. <https://stackabuse.com/introduction-to-image-processing-in-python-with-opencv/>

118. <https://www.naturefocused.com/articles/photography-image-processing-kernel.html>
119. <https://setosa.io/ev/image-kernels/>
120. <https://docs.microsoft.com/en-us/azure/cognitive-services/face/concepts/face-detection>
121. <https://towardsdatascience.com/computer-vision-detecting-objects-using-haar-cascade-classifier-4585472829a9>
122. [https://docs.opencv.org/3.4.3/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/3.4.3/d7/d8b/tutorial_py_face_detection.html)
123. [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)
124. <https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>
125. <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
126. [https://en.wikipedia.org/wiki/Transfer\\_learning](https://en.wikipedia.org/wiki/Transfer_learning)
127. <https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/>
128. <https://nanonets.com/blog/ocr-with-tesseract/>