E-commerce product recommendation system

**ABSTRACT**

Internet usage has increased exponentially during the last ten years. Its expansion has created opportunities for other firms that depend on it to expand and thrive as well. E-commerce is one such route. E-commerce is expanding quickly, and as it does, recommendations are becoming more and more crucial. A recommendation is a pertinent advice made to the user based on their needs and areas of interest. Users may receive recommendations depending on price, location, wish list, basket, searched, and already bought items. Systems that make recommendations increase user engagement, increase sales, and improve the user experience. Both new and returning users will receive product recommendations from our suggested suggestion system. This recommendation system makes product recommendations based on the rating and past purchasing behaviour of long-term customers through model-based collaborative filtering. Additionally, suggestions for new products, popular products, and discounted products will be sent to new users. Recommendations based on recently viewed products, related products, etc. will be sent to current users. In this instance, suggestions are based on the textual clustering analysis of the product description because we are setting up a new e-commerce website and there aren't any user reviews for various products at first. Textual clustering and model-based collaborative filtering will help us target all user types and increase accuracy. E-commerce is becoming more and more popular, and its recommendation system is just the cherry on top.

**Key Words:** E-commerce, recommendation system.

**INTRODUCTION**

Imagine a physical universe in which the same store has hundreds and hundreds of branches. You have a branch in your neighbourhood that is suited to your needs, just as I have one in my neighbourhood that is suited to mine. This would not be feasible in the real world (even with Starbucks locations everywhere), but the trend towards e-commerce, or virtual commerce, has given rise to business models that are unimaginable in the real world. Joe Pine makes the case in his book Mass Customization (Pine, 1993) that businesses must transition from the mass production era of the past, when "standardized products, homogeneous markets, and long product life and development cycles were the rule," to the customized and varied world of the present. Pine contends that creating a single offering is no longer sufficient. Companies must, at the very least, be able to create a variety of products to satisfy the various wants of various customers. E-commerce has given companies the ability to provide customers more options, even though it hasn't always allowed them to manufacture more goods. Customers may select from millions of books at an online store rather than tens of thousands of volumes in a superstore. But more options have also meant that buyers have to analyse more information before they can decide which products best suit their needs.

E-commerce businesses are using mass customization concepts to their online store's display of products rather than the actual products themselves in an effort to combat this information overload (Pine and Gilmore, 1999). Using recommender systems is one method in which mass customization in e-commerce can be accomplished.

E-commerce websites utilize recommender systems to make product recommendations to their clients and give them information to help them make purchasing decisions. Products may be suggested based on the best overall sellers on a website, the customer's demographics, or an examination of the customer's purchasing patterns in the past to forecast future purchasing patterns. In the Personalized product information, in community opinion summaries, product suggestions, and community criticisms are examples of recommendation formats. In general, these suggestion strategies are a component of website personalization since they enable the site to adjust to the needs of individual users. One approach to implement Pine's concepts on the Web is through personalization, to a certain degree.

When mass customization first emerged, it meant physically altering goods and services to suit the requirements of individual customers (Pine, 1993). In the last few years, mass customization has expanded to include many different ways to personalize the customer experience (Pine and Gilmore, 1999). The actual objects, whose functionality and appearance may be customized, as well as the way those things are presented, which can be automatically customized or with the customer's assistance, comprise the consumer experience. Recommender systems, in this more expansive sense, facilitate the personalization of the user experience when it comes to how products are displayed on websites. Recommender systems allow for the construction of a new store that is customised for every customer. Naturally, in the virtual realm, a consumer's display of product options is the only thing that varies—a physical storefront is not involved at all.

While they differ from supply-chain decision-support systems and marketing systems, recommender systems share some similarities. Marketing systems aid in the decision-making process of the marketer by classifying consumers into groups based on marketing segments and classifying products into groups that can be related to these segments. Then, marketing efforts can be launched to persuade customers in various market segments to buy goods from categories that the marketer has chosen. Recommender systems, on the other hand, work directly with customers to assist them in finding goods they will like to buy. Marketing professionals can make judgements about how many products to produce and which warehouses or retail locations to distribute them to with the use of supply-chain decision-support systems.

In order to ensure that customers may acquire the appropriate products, these decision-support systems employ analytical technology to forecast which products, in what quantities, will be purchased at each location. A lot of supply-chain decision-support systems provide answers to aggregate-related queries, such as: how many Minneapolis shoppers will purchase toothpaste in February? Recommender systems provide answers to inquiries regarding specific customers, such as which product the customer would want to purchase at this moment.

Recommender systems comprise tasks that are primarily carried out by hand, like making cross-sell lists by hand, and tasks that are mostly carried out by computers, such collaborative filtering. The latter are what we'll call automatic recommender systems. Specialized data mining technologies called automatic recommender systems are designed to communicate with customers more effectively than with marketers. They are specifically made to capitalize on interactive e-commerce's real-time customization possibilities. As a result, the algorithms place more of an emphasis on just-in-time and real-time learning than on the creation and use of models. Since both manual and automatic recommender systems present a wealth of intriguing concepts regarding how recommendations should be presented to customers, we examine both. The components of recommender systems and their use in e-commerce are introduced in this paper.

This adds five new insights to our knowledge of how recommender systems are used in e-commerce. First, we look at how the development of recommender systems as a tool for E-commerce marketing was made possible by conventional marketing techniques. Second, we propose a taxonomy of recommender applications, which groups them according to the degree of personalization, the technique of generating recommendations, the inputs to the recommender process, and the outputs to the consumer. Third, we look at the trends that show up when the taxonomy is taken into account and pinpoint five types of recommender application models. Currently, recommender systems are generally used in e-commerce according to these five models. Fourth, based on areas of our taxonomy that the current applications have not sufficiently investigated, we outline four areas of future research for novel recommender system applications. Lastly, we address privacy concerns that are changing as more websites start using recommender systems in the appendix.

**Problem Statement**

E-commerce websites and applications have become increasingly popular in recent years, and personalized product recommendations are a key feature of these platforms. However, existing recommendation systems face several challenges, including limited resources, data validity time, and cold start problems. The recommendation system consists of a user model, a recommended model, and a recommendation algorithm. The present recommendation system is ineffective because it doesn’t handle these three main problems limited resource, cold start, data valid time.

The goal of the suggested system is to raise client satisfaction.

**Chapter2**

**Literature Review**

1. One technique utilised in recommendation systems is the collaborative filtering algorithm, which Deng Ailin, Zhu Yangyong, and Shi Bole [1] developed. Conventional algorithms just concentrate on user ratings, disregarding shifts in user interest and the reliability of rating data, both of which have an impact on the suggestion quality of the system. In order to address this issue, a better approach is presented in this study. The premise of the idea is that people who are similar will have similar tastes. Utilize user similarity to identify neighbour’s who share the same interests as the user.

To determine the user's choice, add together the neighbor's scores. Next, the user's top-N items are shown. However, rather than taking user interest into account, the collaborative filtering algorithm's similarity measures focus on the similarity score.

1. Liu Heping and Wang Guoxia [4] suggested a survey on customized recommendation systems. Systems that recommend services are incredibly effective. These find extensive usage in numerous applications. The rapid growth in client base has presented a massive data analytic challenge for service recommendation systems. Collaborative filtering creates a model based on the past behaviour of the client and comparable choices made by people with similar interests. Content-based filtering makes recommendations for more products with similar features based on a variety of an item's attributes.

Systems for hybrid recommendation are created by combining these. Every kind of system has benefits and drawbacks. This study contributes to the selection of optimal algorithm strategies for recommendation, taking into account the system's requirements and input set.

1. Merchandisers began gathering and evaluating data regarding consumer behaviour as soon as they were able to record transaction data. The collection of analysis methods used to extract rules or create models from massive data sets is referred to as data mining. The identification of association rules—relationships between products that suggest a relationship between the purchase of one item and the purchase of another—is one of the most well-known applications of data mining in commerce. These guidelines can assist a merchandiser in setting up products such that, for instance, a customer buying ketchup notices relish close by. If a customer purchases a new charcoal grill today, more advanced temporal data mining may indicate that they will probably purchase a fire extinguisher within the following month.
2. In general, there are two stages to data mining. During the learning phase, the consumer behaviour model (e.g., association rules) is constructed by the data mining system through data analysis. This stage can take a long time and call for the participation of human analysts. Following model construction, the system moves into a use phase where the model can be quickly and simply implemented in real-world customer scenarios. Developing organisational procedures that successfully translate information from the learning phase into practise during the use phase is one of the problems associated with integrating data mining within organisations.
3. Machine learning systems with a focus on product recommendations for use in commerce applications are called automatic recommender systems. Certain recommenders go through two phases: an offline phase where they pick up a model of client behaviour, and an online phase where they use the model in real time. However, the majority of recommenders employ a lazy learning strategy, updating and building the model as they go along while providing recommendations instantly.

**2.5 SOFTWARE ENVIRONMENT**

**Python** is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt. Sources [newsgroup](https://en.wikipedia.org/wiki/Usenet) in 1991, and version 1.0 was released in 1994.

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been back ported to Python 2. But in general, they remain not quite compatible.

Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official [End of Life date of January 1, 2020](https://pythonclock.org/) has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python, it is recommended that you focus on Python 3, as this tutorial will do.

Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe [Monty Python’s Flying Circus](https://en.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus), of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

**2.6 WHY CHOOSE PYTHON**

If you’re going to write programs, there are literally dozens of commonly used languages to choose from. Why choose Python? Here are some of the features that make Python an appealing choice.

**Python is Popular**

Python has been growing in popularity over the last few years. The 2018 [Stack Overflow Developer Survey](https://insights.stackoverflow.com/survey/2018) ranked Python as the 7th most popular and the number one most wanted technology of the year. [World-class software development countries around the globe use Python every single day.](https://realpython.com/world-class-companies-using-python/)

According to [research by Dice](https://insights.dice.com/2016/02/01/whats-hot-and-not-in-tech-skills/) Python is also one of the hottest skills to have and the most popular programming language in the world based on the [Popularity of Programming Language Index](https://pypl.github.io/PYPL.html).

Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you’d like to dig deeper into [Python salary statistics and job opportunities, you can do so here](https://dbader.org/blog/why-learn-python).

**Python is interpreted**

Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer’s processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.

One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

### Python is Free

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes.

A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

### Python is Portable

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

### Python is Simple

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way.

A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect.

Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read.

But It’s Not That Simple For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and [object-oriented programming](https://realpython.com/python3-object-oriented-programming/).

Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.

Python accomplishes what many programming languages don’t: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

## Conclusion

This section gave an overview of the **Python** programming language, including:

* A brief history of the development of Python
* Some reasons why you might select Python as your language of choice

Python is a great option, whether you are a beginning programmer looking to learn the basics, an experienced programmer designing a large application, or anywhere in between. The basics of Python are easily grasped, and yet its capabilities are vast. Proceed to the next section to learn how to acquire and install Python on your computer.

**Python** is an [open source](https://simple.wikipedia.org/wiki/Open_source) [programming language](https://simple.wikipedia.org/wiki/Programming_language) that was made to be easy-to-read and powerful. A [Dutch](https://simple.wikipedia.org/wiki/Netherlands) programmer named [Guido van Rossum](https://simple.wikipedia.org/wiki/Guido_van_Rossum) made Python in 1991. He named it after the television show [Monty Python's Flying Circus](https://simple.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus). Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be [compiled](https://simple.wikipedia.org/wiki/Compiled_language) to run. A program called an [interpreter](https://simple.wikipedia.org/wiki/Interpreter_(computing)) runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like [C](https://simple.wikipedia.org/wiki/C_(programming_language)), because it is not running [machine code](https://simple.wikipedia.org/wiki/Machine_code) directly.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

Python drew inspiration from other programming languages like C, [C++](https://simple.wikipedia.org/wiki/C%2B%2B), [Java](https://simple.wikipedia.org/wiki/Java_(programming_language)), [Perl](https://simple.wikipedia.org/wiki/Perl), and [Lisp](https://simple.wikipedia.org/wiki/LISP).

Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a [semicolon](https://simple.wikipedia.org/wiki/Semicolon) at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language.

**Python use [change / change source]**

Python is used by hundreds of thousands of programmers and is used in many

places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.

Its [standard library](https://simple.wikipedia.org/w/index.php?title=Standard_library&action=edit&redlink=1) is made up of many [functions](https://simple.wikipedia.org/wiki/Computable_function) that come with Python when it is installed. On the [Internet](https://simple.wikipedia.org/wiki/Internet) there are many other [libraries](https://simple.wikipedia.org/w/index.php?title=Library_(computing)&action=edit&redlink=1) available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

* Web development
* Scientific programming
* Desktop [GUIs](https://simple.wikipedia.org/wiki/GUI)
* Network programming
* [Game](https://simple.wikipedia.org/wiki/Video_game) programming

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM:**

Customers find it challenging to locate the products they want using the current E-Commerce recommendation algorithms. Cold start concerns, data validity time constraints, and resource limitations are a few of the problems that current E-Commerce recommendation systems have not given enough thought to. Lack of customer satisfaction, lack of personalized recommendations, inability to resolve cold start issues, improper handling of the restricted resource scenario, improper handling of the data valid time, and decreased efficiency are just a few of the constraints. Collaborative filtering's primary drawback is its reliance on data in order to function. Sparsity and scalability are its two main shortcomings [6]. The inability to distinguish between a good and a bad item based on the information retrieved is a disadvantage of content-based filtering. If a poor item has the same keywords as a good thing, it will also be suggested. Overspecialization and a lack of user information are the other two main drawbacks.

**3.2 PROPOSED SYSTEM:**

The suggested system is a superior e-commerce recommendation system that can provide consumers with useful recommendations that greatly satisfy them. In addition to the numerous advantages for customers, there is also potential for higher trading volume and the resolution of the three previously mentioned issues. The purpose of the proposed system is to put in place a recommendation system that will help customers receive the things they want. As seen in fig. 1, the proposed recommendation system is primarily composed of three models: the user model, the suggested model, and the recommendation algorithm. Customers are more satisfied with the proposed system. The suggested system uses a hybrid algorithm to get around the three issues.

The emergence of several e-commerce websites and applications can be attributed to the significant expansion of the internet over the last ten years. Going online to shop is becoming more and more popular than visiting physical establishments. The primary goal of internet shopping is to make the process of choosing things easier by providing tailored recommendations. Additionally, this is what is anticipated of the e-commerce websites.

**4. FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**4.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**4.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**4.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**5. SYSTEM REQUIREMENTS**

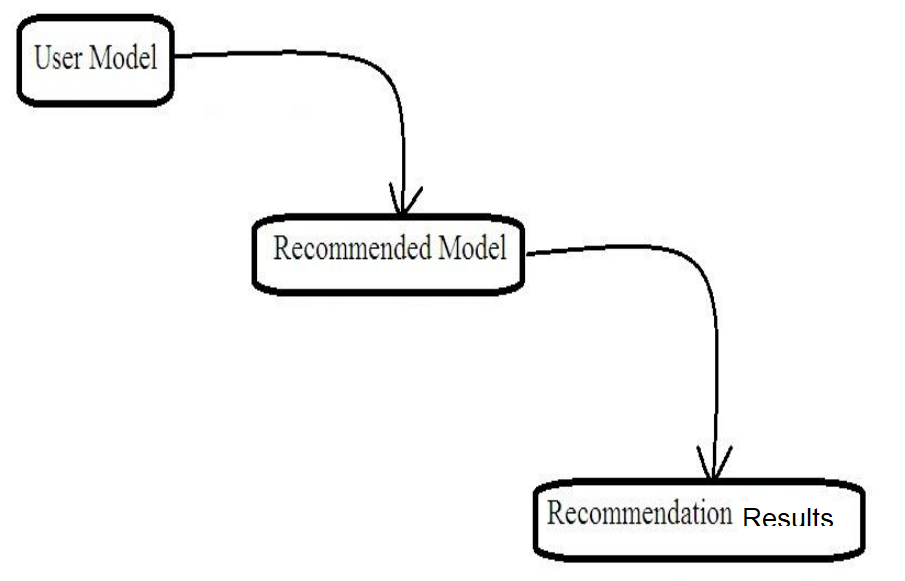
**5.1 HARDWARE REQUIREMENTS:**

* System : Pentium Dual Core.
* Hard Disk : 120 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 4 GB

**5.2 SOFTWARE REQUIREMENTS:**

* Operating system : Windows 10
* Coding Language : python 3.x
* Tool : Python IDLE
* Database : MYSQL
* Server : Streamlit

**6 SYSTEM ARCHITECTURE:**

****

**Fig-1 : proposed model design**

**6.2 DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

**User**

**Unauthorized keypoints**

**Check**

**YES NO**

**Upload Image**

**Pre-Processing**

**Model Load**

**Predicting similar images based on upload image**

**End process**

**6.3 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

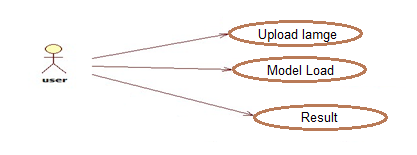
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

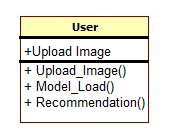
**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



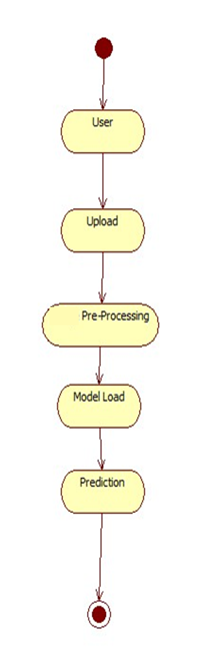
**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

****

**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

****

**7. IMPLEMENTATION:**

Python IDLE was used to implement the desktop application and Jupyter Notebook was used for data preprocessing.Python was the programming language utilised, along with machine learning. The model was constructed using the Random Forest Classification Algorithm with Sklearn.

**SOFTWARE ENVIRONMENT :**

**What is Python :**

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like Kivy, Tkinter, PyQtetc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

**Advantages of Python :-**

Let’s see how Python dominates over other languages.

## 1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

## 2. Extensible

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

## 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

## 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

## 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

## 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

## 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

## 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

## 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

## 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

# Advantages of Python Over Other Languages :

## 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

## 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

## 3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

## Disadvantages of Python:-

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

## History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde&Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voorWiskundeen Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

## What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

## Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

## Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

# How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### (a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### (b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### (c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/), [TensorFlow](https://www.tensorflow.org/), [Scikit-learn](https://scikit-learn.org/stable/), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python) available Free on GeeksforGeeks.

### Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (a) Terminologies of Machine Learning

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### (b) Types of Machine Learning

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### Advantages of Machine learning

#### 1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### 2. No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### 3. Continuous Improvement

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### 5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### Disadvantages of Machine Learning

#### 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

#### 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### 4. High error-susceptibility

[Machine Learning](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

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. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheethere.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [https://www.python.org](https://www.python.org/)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



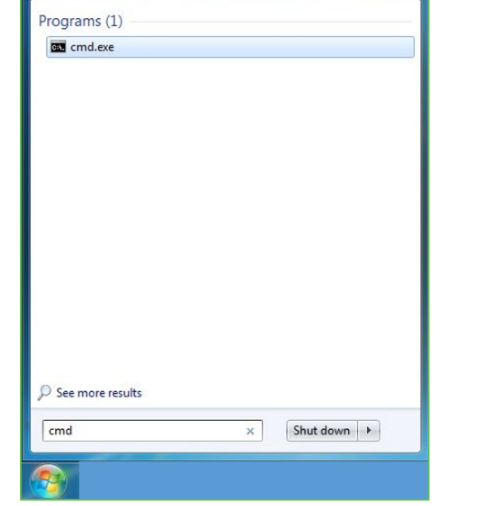
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**

**6.SYSTEM TEST**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTS

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:**All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:**All the test cases mentioned above passed successfully. No defects encountered.

|  |  |  |  |
| --- | --- | --- | --- |
| **TC ID** | **Condition Being Tested** | **Expected Result** | **Result** |
| 1 | Check for null values an duplicate records | Display the duplicate records and null values | passed |
| 2 | TC7 extract features using vectorisation | Display the features of the data | Passed |
| 3 | TC8 check if the data is not empty  And display records | Show error if it is empty else display the data | passed |
| 4 | Display the number of records for training and test | Display the count of training and test  Data | passed |
| 5 | Save the model of the best accuracy graph | Saved in the current path | passed |

**CODE:**

**SCREENSHOTS :**

**CONCLUSION**

The need for personalized user recommendations has grown as e-commerce websites have developed over time. Problems with limited resources, data validity, and cold starts have been unresolved in previous e-commerce recommendation systems; a new approach has been presented here that attempts to address these issues and provide effective recommendations to customers by resolving the cold start issue and limited resource situation. These solutions can then be further integrated into the various e-commerce websites that currently exist. Up until now, we have worked to identify remedies for the shortcomings of the current systems and have a deep understanding of them. adding more fields to address the cold start issue and adding user comments to offer recommendations are potential future work items.

**References and Bibliography**:

[1] A guide to Natural Langugage Processing, Available at

https://en.wikipedia.org/wiki/Natural\_language\_processing

[2] Chatbot definition, Available at https://medium.com/@mg/bot-is-a-

hilariously-over-simplified-buzzword-let-s-fix-that-

f1d63abb8ba7#:~:text=A%20chatterbot%20(also%20known%20as,via%

20auditory%20or%20textual%20methods.

[3] Introduction to Artificial Intelligence Markup Language, Available at

https://www.tutorialspoint.com/aiml/aiml\_introduction.htm

[4] Prof.K.Bala, Mukesh Kumar, SayaliHulawale, SahilPandita,“Chat-Bot

For College Management System Using A.I” International Research

Journal of Engineering and Technology (IRJET) Volume: 04, Issue: 11,

Page no: 2030-2033| Nov 2017.

[5] Porter Stemmer Algorithm, Available at

http://snowball.tartarus.org/algorithms/porter/stemmer.html

[6] Guruswami Hiremath, AishwaryaHajare, PriyankaBhosale,

RasikaNanaware, Dr. K. S. Wagh, “Chatbot for education system”

International Journal of Advance Research, Ideas and Innovations in

Technology (IJARIIT) ISSN: 2454-132X, Volume: 4, Issue: 3, Page no:

37-43|2018.

[7] Amey Tiwari, Rahul Talekar, Prof.S.M.Patil, “College Information Chat

Bot System” International Journal of Engineering Research and General

Science (IJERGS) Volume: 5, Issue: 2, Page no: 131-137| March-April

2017.

[8] K. Jwala, G.N.V.G Sirisha, G.V. Padma Raju, “Developing a Chatbot

using Machine Learning” International Journal of Recent Technology

and Engineering (IJRTE) ISSN: 2277-3878, Volume: 8 Issue: 1S3, Page

no: 89-92| June 2019.

[9] Basics of Natural Language ToolKit, Avalilable at https://www.nltk.org/

[10] ] Naeun Lee, Kirak Kim, Taeseon Yoon, “Implementation of Robot

Journalism by Programming Custombot using Tokenization and Custom

Tagging” International Conference on Advanced Communications

Technology (ICACT) Page no: 566-570| Feb 2017.

[11] Fundamentals of Natural Language Processing - Tokenization,

Lemmatization, Stemming and Sentence Segmentation, Available at

https://colab.research.google.com/github/dair-

ai/notebooks/blob/master/\_notebooks/2020-03-19-

nlp\_basics\_tokenization\_segmentation.ipynb#scrollTo=H7gQFbUxOQt

b

[12] Jazzy spell checker Library, Available at http://jazzy.sourceforge.net/

[13] WordNet Algorithm, Avaliable at https://wordnet.princeton.edu/

[14] Setiaji Bayu, Wibowo Ferry “Chatbot Using a Knowledge in Database:

Human-to-Machine Conversation Modeling” 7th International

Conference on Intelligent Systems, Modelling and Simulation (ISMS)

Page no: 72-77| Jan 2016. DOI: 10.1109/ISMS.2016.53

[1] A guide to Natural Langugage Processing, Available at

https://en.wikipedia.org/wiki/Natural\_language\_processing

[2] Chatbot definition, Available at https://medium.com/@mg/bot-is-a-

hilariously-over-simplified-buzzword-let-s-fix-that-

f1d63abb8ba7#:~:text=A%20chatterbot%20(also%20known%20as,via%

20auditory%20or%20textual%20methods.

[3] Introduction to Artificial Intelligence Markup Language, Available at

https://www.tutorialspoint.com/aiml/aiml\_introduction.htm

[4] Prof.K.Bala, Mukesh Kumar, SayaliHulawale, SahilPandita,“Chat-Bot

For College Management System Using A.I” International Research

Journal of Engineering and Technology (IRJET) Volume: 04, Issue: 11,

Page no: 2030-2033| Nov 2017.

[5] Porter Stemmer Algorithm, Available at

http://snowball.tartarus.org/algorithms/porter/stemmer.html

[6] Guruswami Hiremath, AishwaryaHajare, PriyankaBhosale,

RasikaNanaware, Dr. K. S. Wagh, “Chatbot for education system”

International Journal of Advance Research, Ideas and Innovations in

Technology (IJARIIT) ISSN: 2454-132X, Volume: 4, Issue: 3, Page no:

37-43|2018.

[7] Amey Tiwari, Rahul Talekar, Prof.S.M.Patil, “College Information Chat

Bot System” International Journal of Engineering Research and General

Science (IJERGS) Volume: 5, Issue: 2, Page no: 131-137| March-April

2017.

[8] K. Jwala, G.N.V.G Sirisha, G.V. Padma Raju, “Developing a Chatbot

using Machine Learning” International Journal of Recent Technology

and Engineering (IJRTE) ISSN: 2277-3878, Volume: 8 Issue: 1S3, Page

no: 89-92| June 2019.

[9] Basics of Natural Language ToolKit, Avalilable at https://www.nltk.org/

[10] ] Naeun Lee, Kirak Kim, Taeseon Yoon, “Implementation of Robot

Journalism by Programming Custombot using Tokenization and Custom

Tagging” International Conference on Advanced Communications

Technology (ICACT) Page no: 566-570| Feb 2017.

[11] Fundamentals of Natural Language Processing - Tokenization,

Lemmatization, Stemming and Sentence Segmentation, Available at

https://colab.research.google.com/github/dair-

ai/notebooks/blob/master/\_notebooks/2020-03-19-

nlp\_basics\_tokenization\_segmentation.ipynb#scrollTo=H7gQFbUxOQt

b

[12] Jazzy spell checker Library, Available at http://jazzy.sourceforge.net/

[13] WordNet Algorithm, Avaliable at https://wordnet.princeton.edu/

[14] Setiaji Bayu, Wibowo Ferry “Chatbot Using a Knowledge in Database:

Human-to-Machine Conversation Modeling” 7th International

Conference on Intelligent Systems, Modelling and Simulation (ISMS)

Page no: 72-77| Jan 2016. DOI: 10.1109/ISMS.2016.53

[1] Deng Ailin, Zhu Yangyong, Shi Bole. A Collaborative Filtering Recommendation Algorithm Based on Item Rating Prediction[J]. Journal of Software.2003(09) .

[2] G. Adomavicius, A. Tuzhilin. Toward the next generation of recommender systems:A survey of the Stateof the-art and possible extensions[J]. IEEE Transactions on Knowledge and Data Engineering,2005,17(06):734- 749.DOI:10.1002/pon.1586.

[3] Sun Dongting, He Tao, Zhang Fuhai. Survey of Coldstart Problem in Collaborative Filtering Recommender System[J].Computer and Modernization,2012,(5)

[4] Wang Guoxia, Liu Heping. Survey of personalized recommendation system[J]. Computer Engineering and Applications,2012,48(7):66- 76.DOI:10.3778/j.issn.10028331.2012.07.018.

[5] Jiang Shaowei. Recommendation Models Based on Competitive Relationship[D]. Beijing University of Posts and Telecommunications,2014.

[6]

G. Adoma

vicius and A. Tuzhilin, "Toward the next generation of

recommender systems: a survey of the state-of-the-

art and possible

extensions," Knowledge and Data Engineering, IEEE Transactions,

vol.

17, no. 6, pp. 734-749, 2005.

[7]

M. Claypool, P. Le, M. Wased

a and D. Brown, "Implicit Interest

Indicators," in ACM, New York, 2001.

[8]

A. N. Regi and R. Sandra, "A Survey on Recommendation Techniques in

E-Commerce," vol. 2, no. 12, 2013.

[9]

R. J. Brachman, T. Khabaza, W. Kloesgen, G. Piatetsky-

Shapiro and E.

Simoudis, "Mining business databases," in ACM, New York, 1996.

[10]

B. Schafer, J. Konstan and J. Riedl, "E-

Commerce Recommendation

Applications," in Applications of Data Mining to Electronic Commerce

,

Minneapolis, Springer US, 2001, pp. 115-153.

[11]

R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association

Rules," in 20th International Conference on Very Large Databases

,

Santiago, 1994.

[6]

G. Adoma

vicius and A. Tuzhilin, "Toward the next generation of

recommender systems: a survey of the state-of-the-

art and possible

extensions," Knowledge and Data Engineering, IEEE Transactions,

vol.

17, no. 6, pp. 734-749, 2005.

[7]

M. Claypool, P. Le, M. Wased

a and D. Brown, "Implicit Interest

Indicators," in ACM, New York, 2001.

[8]

A. N. Regi and R. Sandra, "A Survey on Recommendation Techniques in

E-Commerce," vol. 2, no. 12, 2013.

[9]

R. J. Brachman, T. Khabaza, W. Kloesgen, G. Piatetsky-

Shapiro and E.

Simoudis, "Mining business databases," in ACM, New York, 1996.

[10]

B. Schafer, J. Konstan and J. Riedl, "E-

Commerce Recommendation

Applications," in Applications of Data Mining to Electronic Commerce

,

Minneapolis, Springer US, 2001, pp. 115-153.

[11]

R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association

Rules," in 20th International Conference on Very Large Databases

,

Santiago, 1994.

[6] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," Knowledge and Data Engineering, IEEE Transactions, vol. 17, no. 6, pp. 734-749, 2005.

[7] M. Claypool, P. Le, M. Waseda and D. Brown, "Implicit Interest Indicators," in ACM, New York, 2001.

[8] A. N. Regi and R. Sandra, "A Survey on Recommendation Techniques in E-Commerce," vol. 2, no. 12, 2013.

[9] R. J. Brachman, T. Khabaza, W. Kloesgen, G. Piatetsky-Shapiro and E. Simoudis, "Mining business databases," in ACM, New York, 1996.

[10] B. Schafer, J. Konstan and J. Riedl, "E-Commerce Recommendation Applications," in Applications of Data Mining to Electronic Commerce, Minneapolis, Springer US, 2001, pp. 115-153.

[11] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in 20th International Conference on Very Large Databases, Santiago, 1994.

