

**National Institute of Technical  
Teachers Training and Research  
(NITTTR, CHANDIGARH)**



**MANNUAL ON  
RECOMMENDER SYSTEM**

# CONTENTS

1. INTRODUCTION .....	2
2. WHAT IS RECOMMENDER SYSTEM? .....	4
3. WHY RECOMMENDARER SYSTEM? .....	6
4. TYPES OF RECOMMENDERSYSTEM? .....	7
5. CONTENT BASED FILTERING .....	7
6. COLLABORATIVE BASED FILTERING .....	9
7. HYBRI BASED FILTERING .....	12
8. PHASES OF RECOMMENDER SYSTEM .....	14
9. REQUIREMENTS .....	16
10. INSTALLATION OF ANACONDA.....	17
11. BASICS ON PYTHON .....	31
12. BASIC ON NUMPY .....	37
13. BASIC ON PANDAS .....	39
14. BASIC ON MATPLOTLIB .....	40
15. DIFFERENT STEPS TO CONSTRUCT BASIC RECOMMENDER SYSTEM .....	42
16.BENEFITS OF RECOMMENDER SYSSTEM.....	56
17. CONCLUSION.....	57
18. ASSIGNMENT.....	58

# INTRODUCTION

An abundant amount of information is created and delivered over electronic media. Users risk becoming overwhelmed by the flow of information, and they lack adequate tools to help them manage the situation. Information filtering (IF) is one of the methods that is rapidly evolving to manage large information flows. The aim of IF is to expose users to only information that is relevant to them. Many IF systems have been developed in recent years for various application domains. Some examples of filtering applications are: filters for search results on the internet that are employed in the Internet software, personal e-mail filters based on personal profiles, list servers or newsgroups filters for groups or individuals, browser filters that block non-valuable information, filters designed to give children access them only to suitable pages, filters for e-commerce applications that address products and promotions to potential customers only, and many more. It deals with the delivering the information which the user is going to like or they feel useful. The information filtering system assist the user and provide the relevant information from the data source.

In the past, people used to shop in a physical store, in which the items available are limited. For instance, the number of movies that can be placed in a Blockbuster store depends on the size of that store. By contrast, nowadays, the Internet allows people to access abundant resources online. On the Internet, where the number of choices is overwhelming, there is need to filter, prioritize and efficiently deliver relevant information in order to alleviate the problem of information overload, which has created a potential problem to many Internet users. Recommender systems solve this problem by searching through large volume of dynamically generated information to provide users with

personalized content and services .Netflix, for example, has an enormous collection of movies. Although the amount of available information increased, a new problem arose as people had a hard time selecting the items they actually want to see. This is where the recommender system comes in.

## History

Before internet, there are already several methods of filtering information; for instance, governments may control and restrict the flow of information in a given country by means of formal or informal censorship. Let's talk about information filters if we refer to newspaper editors and journalists when they provide a service that selects the most valuable information for their clients, readers of books, magazines, newspapers, radio listeners and television viewers. This filtering operation is also present in schools and universities where there is a selection of information to provide assistance based on academic criteria to customers of this service, the students. With the advent of the Internet it is possible that anyone can publish anything he wishes at a low-cost. In this way, it increases considerably the less useful information and consequently the quality information is disseminated. With this problem, it began to devise new filtering with which we can get the information required for each specific topic to easily and efficiently.

# What are recommender systems?

The sudden explosion in the amount of digital information and the number of user of Internet have created a potential challenge of information overload which hinders timely access to items of interest. Information retrieval systems, such as Google, DevilFinder and AltaVista have partially solved this problem but prioritization and personalization of information were absent. This has increased the demand for recommender systems.

Recommender systems aim to predict users' interests and recommend product items that quite likely are interesting for them. Data required for recommender systems stems from explicit user ratings after watching a movie or listening to a song, from implicit search engine queries and purchase histories, or from other knowledge about the users/items themselves.

Recommender system is defined as a decision making strategy for users under complex information environments. From the perspective of E-commerce as a tool that helps users search through records of knowledge which is related to users' interest and preference. It can also be defined as a means of assisting and augmenting the social process of using recommendations of others to make choices when there is no sufficient personal knowledge or experience of the alternatives. Handle the problem of information overload that users normally encounter by providing them with personalized, exclusive content and service recommendations.

# AREAS WHERE RECOMMENDER SYSTEM USED.

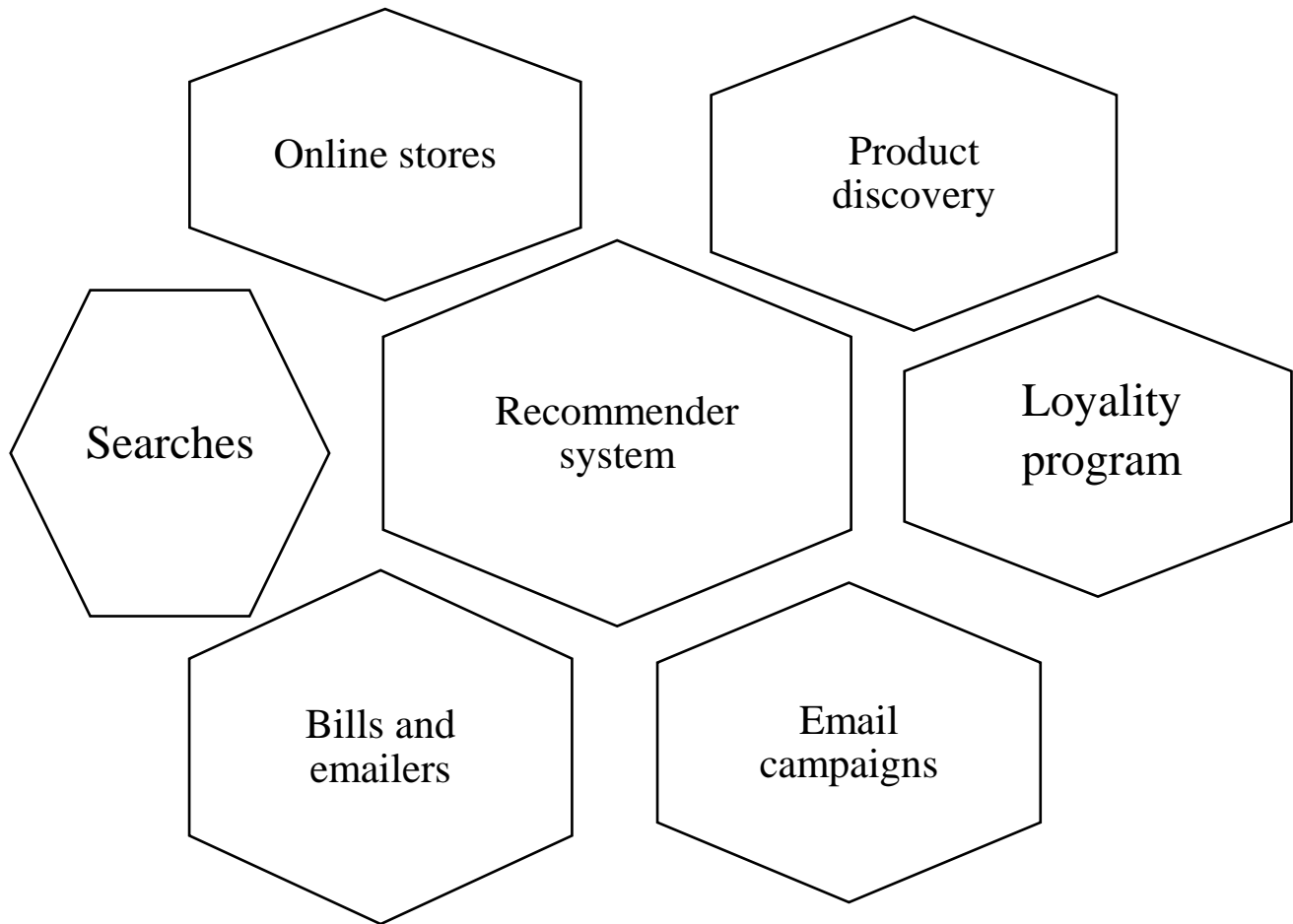


Fig -1

# Why do we need recommender systems?

1. Companies using recommender systems focus on increasing sales as a result of very personalized offers and an enhanced customer experience.
2. Recommendations typically speed up searches and make it easier for users to access content they're interested in.
3. The user starts to feel known and understood and is more likely to buy additional products or consume more content. By knowing what a user wants, the company gains competitive advantage and the threat of losing a customer to a competitor decreases.
4. Recommender systems are information filtering systems that deal with the problem of information overload.
5. It has the ability to predict whether a particular user would prefer an item or not based on the user's profile.
6. Recommender systems are beneficial to both service providers and users. They reduce transaction costs of finding and selecting items in an online shopping environment.
7. Recommendation systems have also proved to improve decision making process and quality.
8. In e-commerce setting, recommender systems enhance revenues, for the fact that they are effective means of selling more products.
9. In scientific libraries, recommender systems support users by allowing them to move beyond catalog searches.

# TYPES OF RECOMMENDER SYSTEM

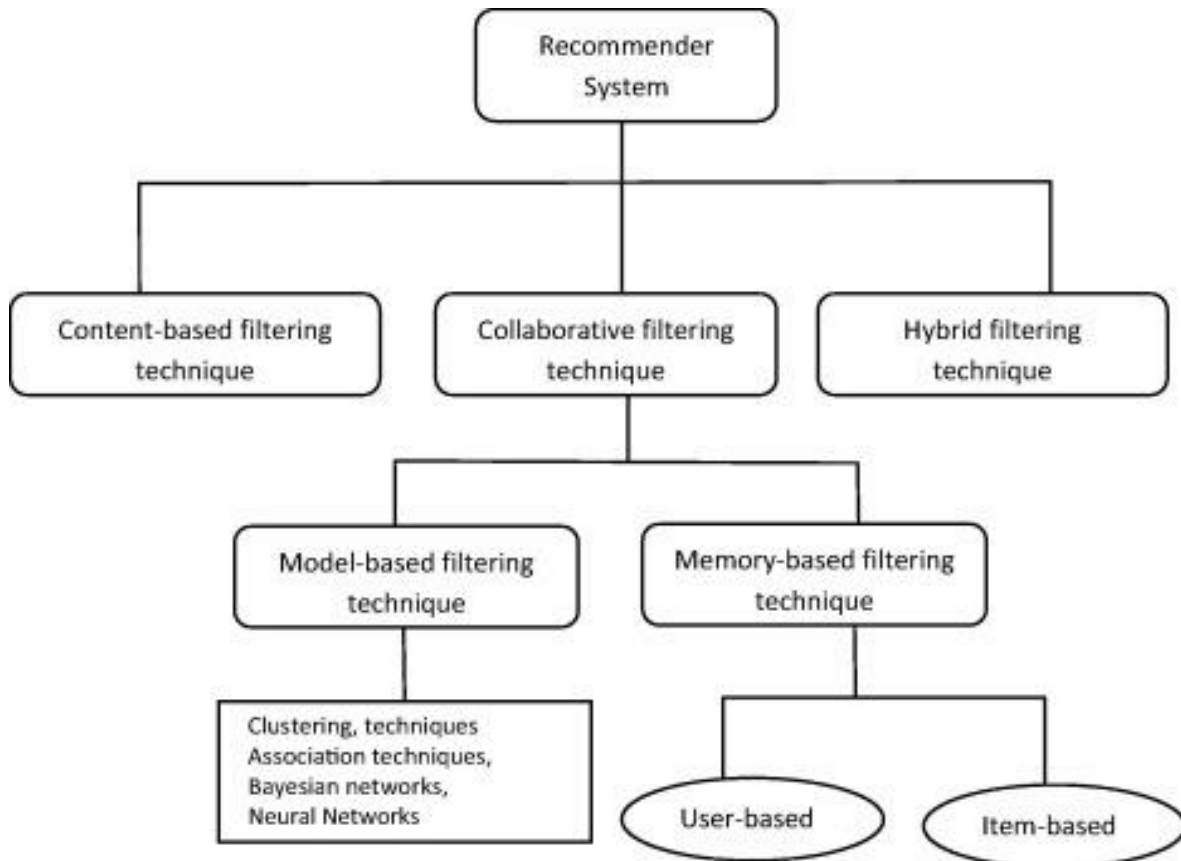


Fig-2

## Content Based Filtering:-

These systems make recommendations using a user's item and profile features. They hypothesize that if a user was interested in an item in the past, they will once again be interested in it in the future. Similar items are usually grouped based on their features. User profiles are constructed using historical interactions or by explicitly asking users about their



interests. There are other systems, not considered purely content-based, which utilize user personal and social data.

Content-based technique is a domain-dependent algorithm and it emphasizes more on the analysis of the attributes of items in order to generate predictions. When documents such as web pages, publications and news are to be recommended, content-based filtering technique is the most successful. In content-based filtering technique, recommendation is made based on the user profiles using features extracted from the content of the items the user has evaluated in the past.

In order to generate meaningful recommendations we use Vector Space Model such as Term Frequency Inverse Document Frequency (TF/IDF) or Probabilistic models such as Naïve Bayes Classifier, Decision Trees or Neural Networks to model the relationship between different documents within a corpus. These techniques make recommendations by learning the underlying model with either statistical analysis or machine learning techniques.

## Pros and Cons of Content-Based Filtering Techniques

A common problem is that new users lack a defined profile unless they are explicitly asked for information. Nevertheless, it is relatively simple to add new items to the system. We just need to ensure that we assign them a group according to their features. They have the ability to recommend new items even if there are no ratings provided by users. So even if the database does not contain user preferences, recommendation accuracy is not affected. Also, if the user preferences change, it has the capacity to adjust its recommendations in a short span of time. They can manage situations where different users do not share the same items, but only identical items according to their intrinsic features. Users can get recommendations without sharing their profile, and this ensures privacy.

Content based filtering techniques are dependent on items' metadata. That is, they require rich description of items and very well organized user profile before recommendation can be made to users. This is called limited content analysis. So, the effectiveness of CBF depends on the availability of descriptive data.

## Collaborative filtering:-

Collaborative filtering technique is the most mature and the most commonly implemented. Collaborative filtering recommends items by identifying other users with similar taste; it uses their opinion to recommend items to the active user. Collaborative recommender systems have been implemented in different application areas. GroupLens is a news-based architecture which employed collaborative methods in assisting users to locate articles from massive news database. Collaborative filtering is currently one of the most frequently used approaches and usually provides better results than content-based recommendations. Some examples of this are found in the recommendation systems of YouTube, Netflix, and Spotify.

Collaborative filtering is a domain-independent prediction technique for content that cannot easily and adequately be described by metadata such as movies and music. Collaborative filtering technique works by building a database (user-item matrix) of preferences for items by users. It then matches users with relevant interest and preferences by calculating similarities between their profiles to make recommendations. Such users build a group called neighborhood. A user gets recommendations to those items that he has not rated before but that were already positively rated by users in his neighborhood.

The system uses collaborative filtering method to overcome scalability issue by generating a table of similar items offline through the use of item-to-item matrix. The system then recommends other products which are similar online according to the users' purchase history.

There are two types of methods to achieve this goal: *memory-based* and *model-based*.

### **Memory-based:-**

There are two approaches: the first one identifies clusters of users and utilizes the interactions of one specific user to predict the interactions of other similar users. The second approach identifies clusters of items that have been rated by user A and utilizes them to predict the interaction of user A with a different but similar item B. These methods usually encounter major problems with large sparse matrices, since the number of user-item interactions can be too low for generating high quality clusters.

### **Model-based:-**

These methods are based on machine learning and data mining techniques. The goal is to train models to be able to make predictions. For example, we could use existing user-item interactions to train a model to predict the top-5 items that a user might like the most. One advantage of these methods is that they are able to recommend a larger number of items to a larger number of users, compared to other methods like memory-based.

## **Pros and Cons of Collaborative Filtering Techniques**

Collaborative Filtering has some major advantages over CBF in that it can perform in domains where there is not much content associated with items and where content is difficult for a computer system to analyze (such as opinions and ideal). Also, CF technique has the ability to provide serendipitous recommendations, which means that it can recommend items that are relevant to the user even without the content being in the user's profile.

Issues with collaborative filtering systems are as defined:-

### Cold-start problem

This refers to a situation where a recommender does not have adequate information about a user or an item in order to make relevant predictions .This is one of the major problems that reduce the performance of recommendation system. The profile of such new user or item will be empty since he has not rated any item; hence, his taste is not known to the system.

### Data sparsity problem

This is the problem that occurs as a result of lack of enough information, that is, when only a few of the total number of items available in a database are rated by users. This always leads to a sparse user-item matrix, inability to locate successful neighbors and finally, the generation of weak recommendations.

### Scalability

This is another problem associated with recommendation algorithms because computation normally grows linearly with the number of users and items .A recommendation technique that is efficient when the number of dataset is limited may be unable to generate satisfactory number of recommendations when the volume of dataset is increased. Thus, it is crucial to apply recommendation techniques which are capable of scaling up in a successful manner as the number of dataset in a database increases.

### Synonymy

Synonymy is the tendency of very similar items to have different names or entries. Most recommender systems find it difficult to make distinction between closely related items such as the difference between e.g. baby

wear and baby cloth. Collaborative Filtering systems usually find no match between the two terms to be able to compute their similarity.

## Hybrid Filtering:-

Hybrid filtering technique combines different recommendation techniques in order to gain better system optimization to avoid some limitations and problems of pure recommendation systems. The idea behind hybrid techniques is that a combination of algorithms will provide more accurate and effective recommendations than a single algorithm as the disadvantages of one algorithm can be overcome by another algorithm. The combination of approaches can be done in any of the following ways: separate implementation of algorithms and combining the result, utilizing some content-based filtering in collaborative approach, utilizing some collaborative filtering in content-based approach, creating a unified recommendation system that brings together both approaches.

Different types hybrid filtering are

### Weighted hybridization

Weighted hybridization combines the results of different recommenders to generate a recommendation list or prediction by integrating the scores from each of the techniques in use by a linear formula. They are given equal weights at first, but weights are adjusted as predictions are confirmed or otherwise. The benefit of a weighted hybrid is that all the recommender system's strengths are utilized during the recommendation process in a straightforward way.

### Switching hybridization

The system swaps to one of the recommendation techniques according to a heuristic reflecting the recommender ability to produce a good rating.

The switching hybrid has the ability to avoid problems specific to one method e.g. the new user problem of content-based recommender, by switching to a collaborative recommendation system.

### Cascade hybridization

The cascade hybridization technique applies an iterative refinement process in constructing an order of preference among different items. The recommendations of one technique are refined by another recommendation technique. The first recommendation technique outputs a coarse list of recommendations which is in turn refined by the next recommendation technique. The hybridization technique is very efficient and tolerant to noise due to the coarse-to-finer nature of the iteration.

### Mixed hybridization

Mixed hybrids combine recommendation results of different recommendation techniques at the same time instead of having just one recommendation per item. Each item has multiple recommendations associated with it from different recommendation techniques. In mixed hybridization, the individual performances do not always affect the general performance of a local region.

### Feature-combination

The features produced by a specific recommendation technique are fed into another recommendation technique. For example, the rating of similar users which is a feature of collaborative filtering is used in a case-based reasoning recommendation technique as one of the features to determine the similarity between items.

### Feature-augmentation

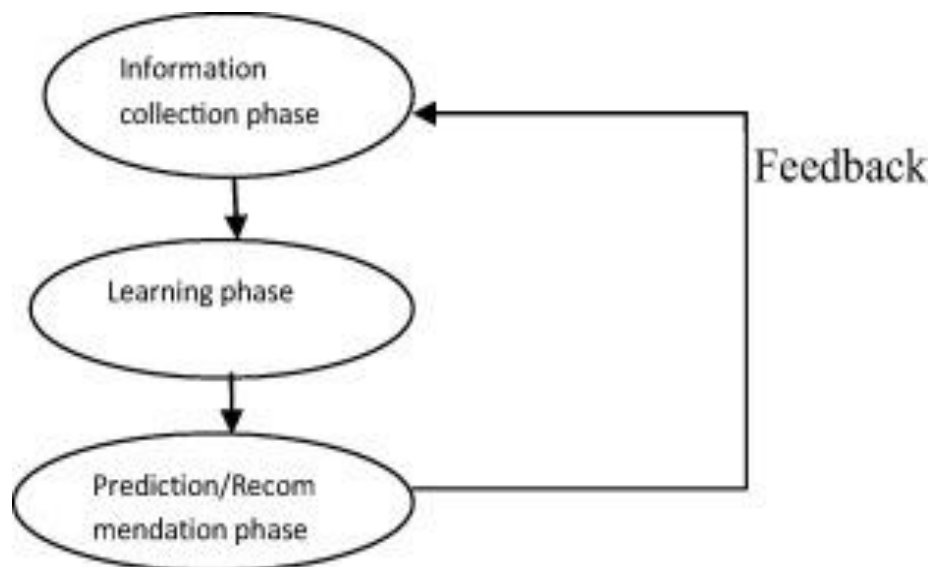
The technique makes use of the ratings and other information produced by the previous recommender and it also requires additional functionality from the recommender systems. Feature-augmentation hybrids are

superior to feature-combination methods in that they add a small number of features to the primary recommender.

## Meta-level

The internal model generated by one recommendation technique is used as input for another. The model generated is always richer in information when compared to a single rating. Meta-level hybrids are able to solve the sparsity problem of collaborative filtering techniques by using the entire model learned by the first technique as input for the second technique.

# Phases of Recommender System



## Information Collection Phase:-

This collects relevant information of users to generate a user profile or model for the prediction tasks including user's attribute, behaviors or content of the resources the user accesses. A recommendation agent cannot function accurately until the user profile/model has been well constructed. In E-learning platform, a user profile is a collection of personal information associated with a specific user. This information

includes cognitive skills, intellectual abilities, learning styles, interest, preferences and interaction with the system. The user profile is normally used to retrieve the needed information to build up a model of the user.

### **Explicit Feedback:-**

Explicit feedback requires more effort from user, it is still seen as providing more reliable data, since it does not involve extracting preferences from actions, and it also provides transparency into the recommendation process that results in a slightly higher perceived recommendation quality and more confidence in the recommendations.

### **Implicit feedback:-**

Implicit feedback reduces the burden on users by inferring their user's preferences from their behavior with the system. The method though does not require effort from the user, but it is less accurate.

### **Learning phase**

It applies a learning algorithm to filter and exploit the user's features from the feedback gathered in information collection phase.

### **Prediction/recommendation phase**

It recommends or predicts what kind of items the user may prefer. This can be made either directly based on the dataset collected in information collection phase which could be memory based or model based or through the system's observed activities of the user.



# What Are The Prerequisites For Building A Recommender System?

Data is the single most important asset. Essentially, you need to know some details about your users and items. If metadata is all you have available, you can start with content-based approaches. If you have a large number of user interactions, you can experiment with more powerful collaborative filtering. The larger the data set in your possession, the better your systems will work.

## What is metadata?

Metadata is "data that provides information about other data". In other words, it is "data about data." Many distinct types of metadata exist, including descriptive metadata, structural metadata, administrative metadata reference metadata and statistical metadata. Descriptive metadata is descriptive information about a resource. It is used for discovery and identification. It includes elements such as title, abstract, author, and keywords.

- Structural metadata is metadata about containers of data and indicates how compound objects are put together, for example, how pages are ordered to form chapters. It describes the types, versions, relationships and other characteristics of digital materials.
- Administrative metadata is information to help manage a resource, like resource type, permissions, and when and how it was created.
- Reference metadata is information about the contents and quality of statistical data.
- Statistical metadata, also called process data, may describe processes that collect, process, or produce statistical data

# STEP 1: - INSTALLATION OF ANACONDA

1. To download the installer: -

<https://www.anaconda.com/distribution/>

CLICK

We get the following: -

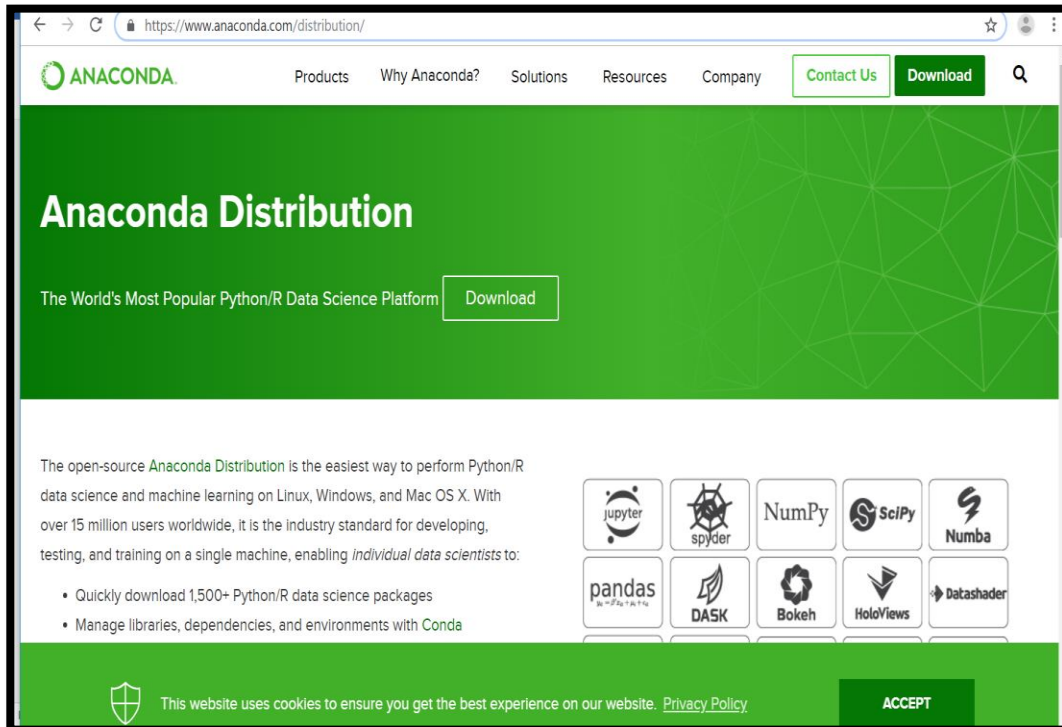


Fig-3

a. We now need to download according to your operating system windows 64/32bits or version 2 or 3 but recommended version 3.

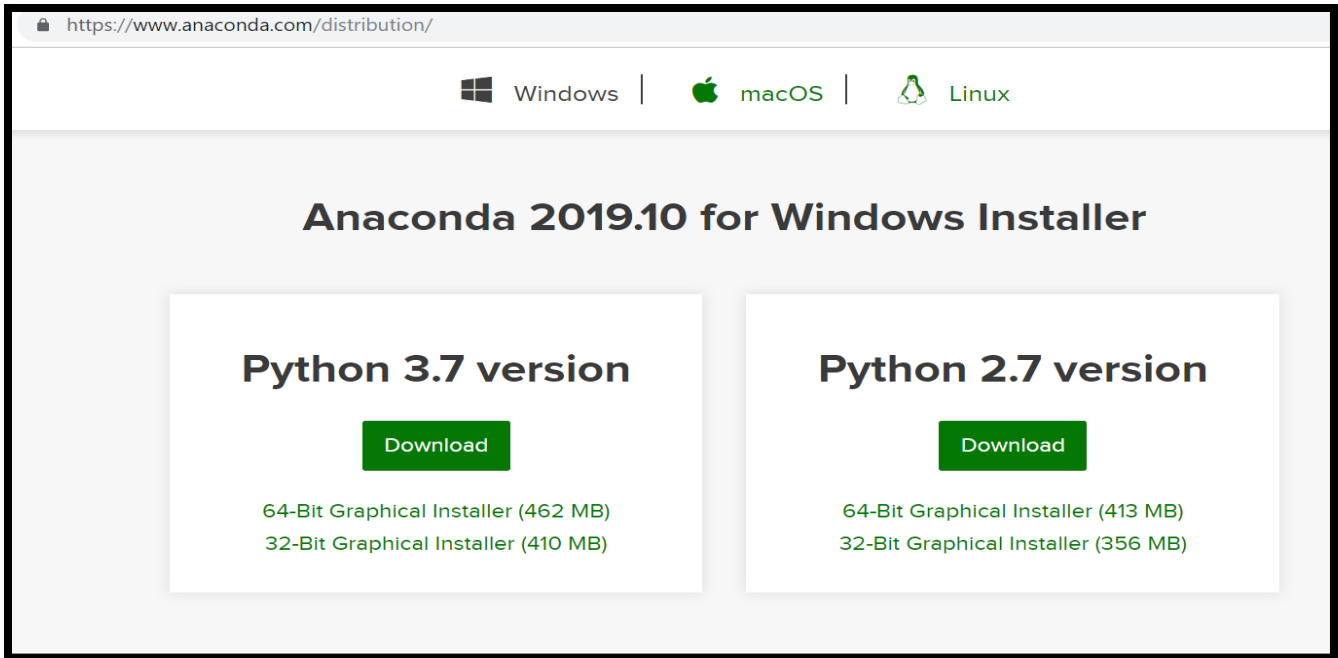


Fig - 4

b. On clicking on the download we get an .exe file. We click on the .exe file we get the following: -



Fig - 5

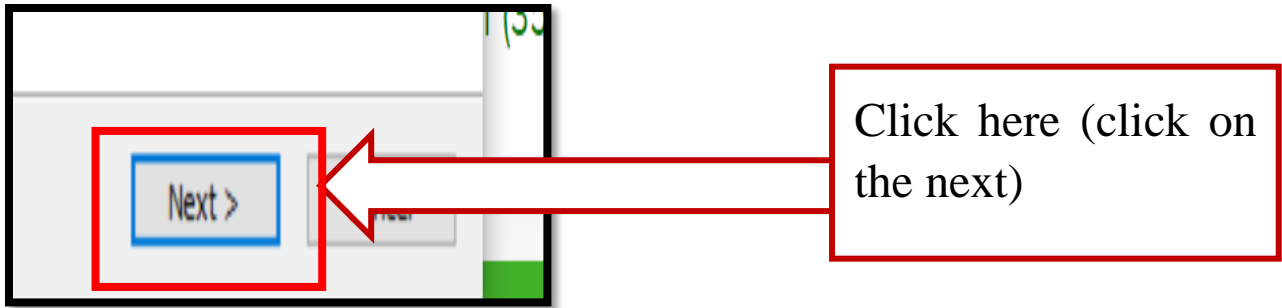


Fig - 6

c. Now click on next and then on "I agree" followed by just me option a next option. Choosing different position. Click on next.

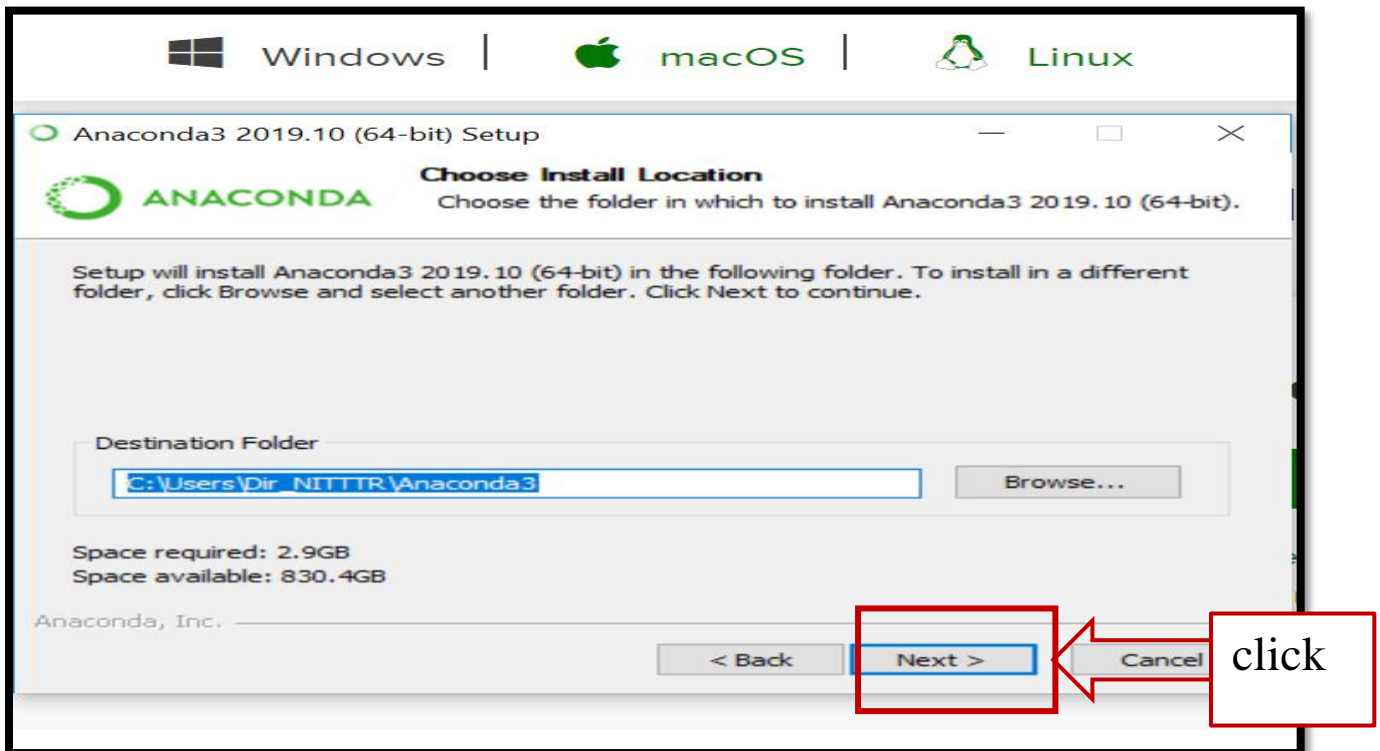


Fig - 7

d. Clicking on next we get the following: -

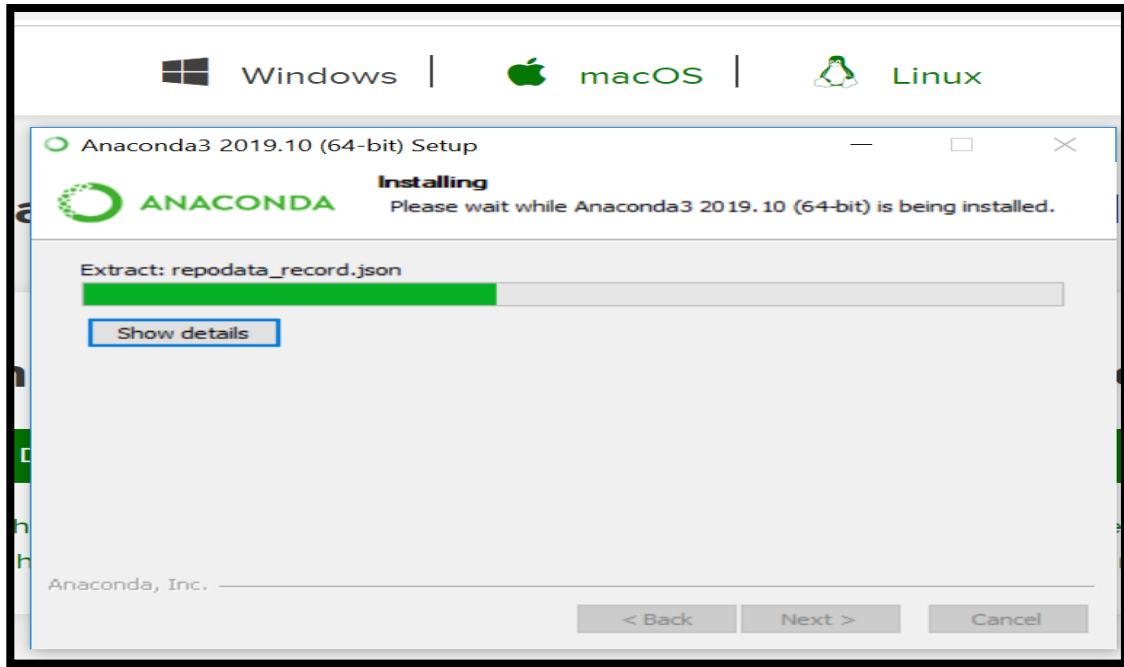


Fig – 8

Note: - It may take some time.

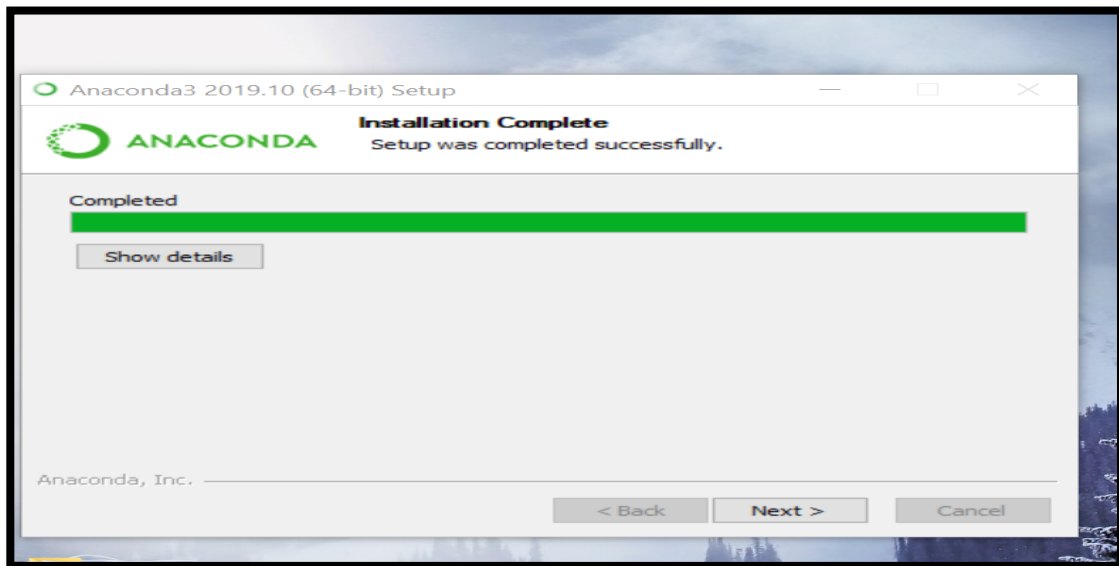


Fig - 9

Then click on next, again next and the finish  
e. The go to windows start menu. Click on the anaconda prompt.

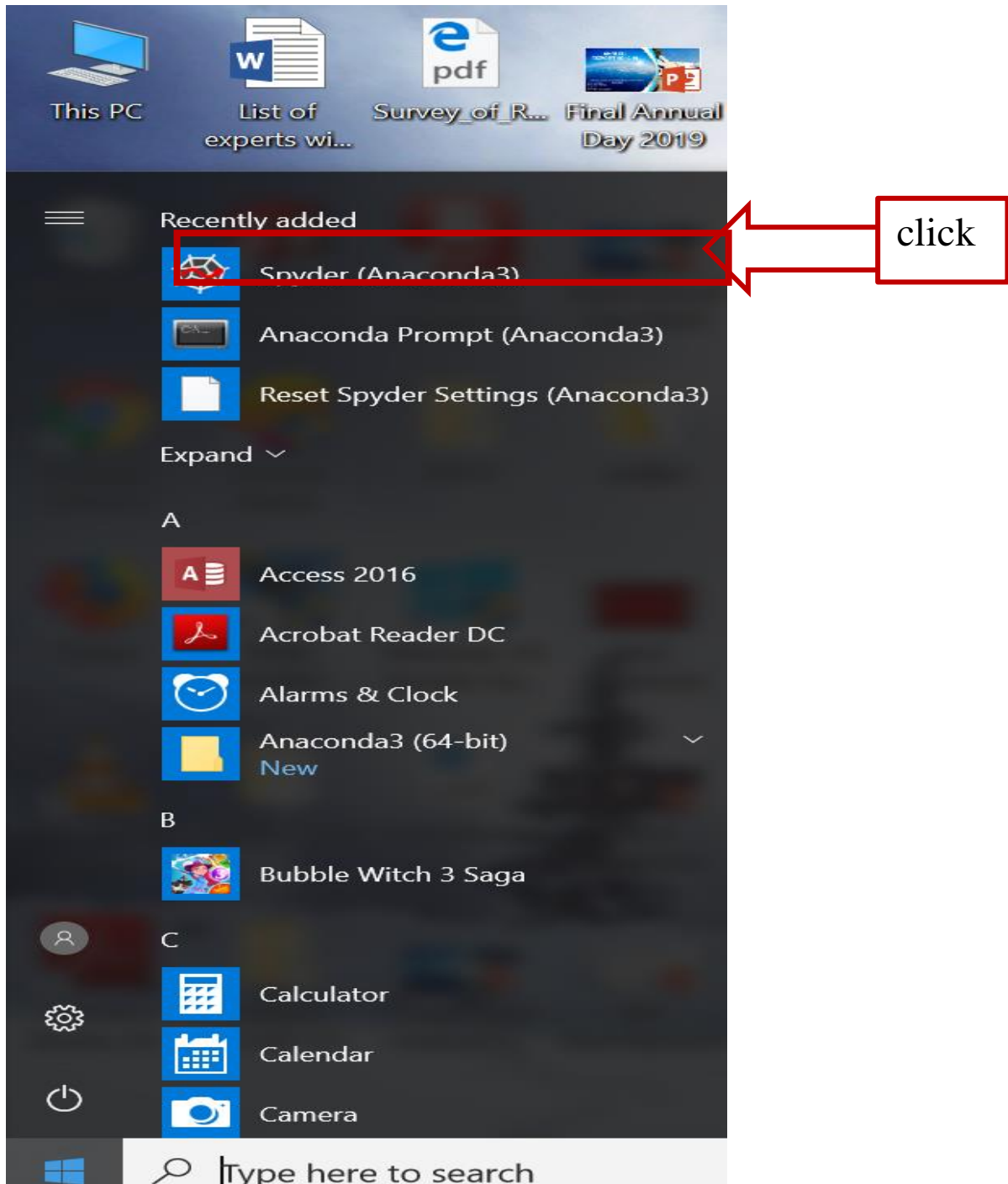


Fig -10

f. Then at Anaconda prompt do the following-

g. Write the following commands: - > python

Press  
Enter

```
python

(base) C:\Users\Dir_NITTR>python
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Fig - 11

h. Type `>import this`. It will import all the function required.

```
python

(base) C:\Users\Dir_NITTR>python
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>> █
```

Fig -12

Then type “exit ()”

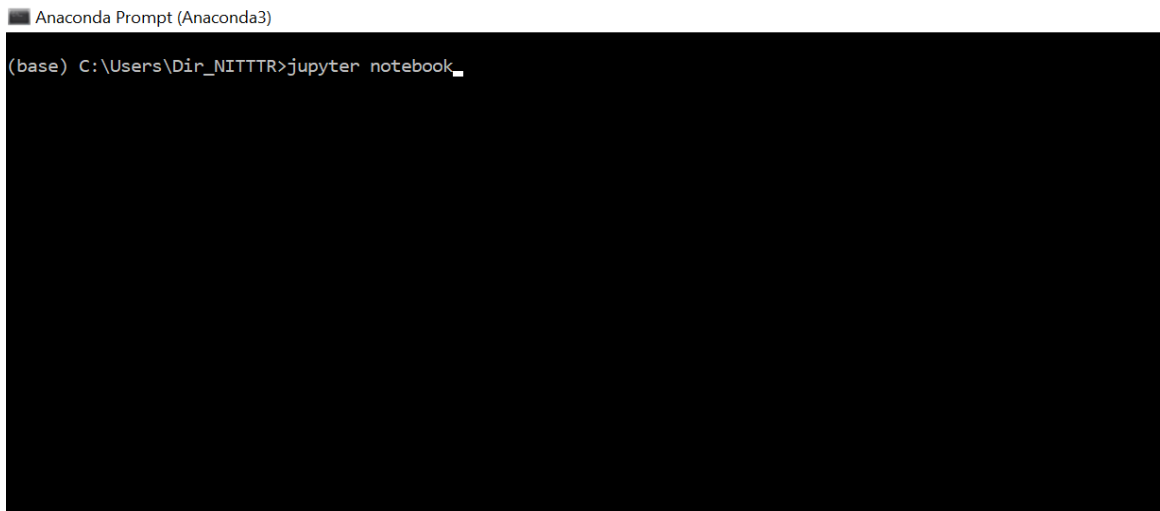
## Step 2: - Installing Jupyter On Windows

### 1. Installing Jupyter on Windows using the Anaconda Prompt

a. To install Jupyter on Windows, open the **Anaconda Prompt** and type:

```
>conda install Jupyter
```

Type ‘y’ for yes when prompted. Once Jupyter is installed, type the command below into the **Anaconda Prompt** to open the Jupyter notebook file browser and start using Jupyter notebooks.

A screenshot of the Anaconda Prompt terminal window. The title bar reads "Anaconda Prompt (Anaconda3)". The terminal shows the prompt "(base) C:\Users\Dir\_NITTR>" followed by the command "jupyter notebook\_" being entered. The rest of the terminal area is black.

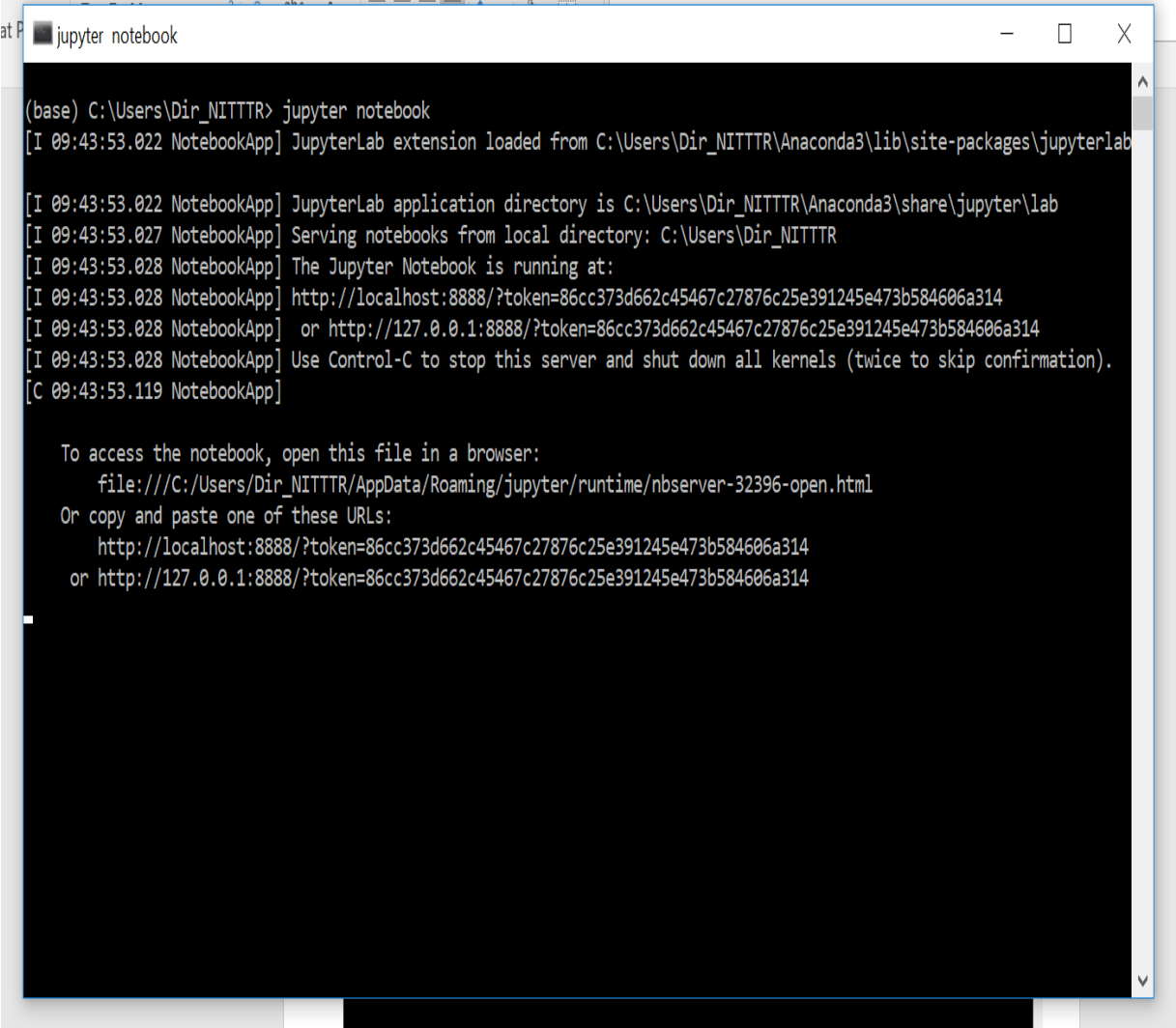
```
Anaconda Prompt (Anaconda3)
(base) C:\Users\Dir_NITTR>jupyter notebook_
```

Fig – 13

Or you can directly type “jupyter notebook “ in the anaconda command prompt.as shown below. (As some system may go directly)



b. Next we get the following: -

The image shows a terminal window titled 'jupyter notebook'. The prompt is '(base) C:\Users\Dir\_NITTTR> jupyter notebook'. The output consists of several lines of log messages: '[I 09:43:53.022 NotebookApp] JupyterLab extension loaded from C:\Users\Dir\_NITTTR\Anaconda3\lib\site-packages\jupyterlab...', '[I 09:43:53.022 NotebookApp] JupyterLab application directory is C:\Users\Dir\_NITTTR\Anaconda3\share\jupyter\lab', '[I 09:43:53.027 NotebookApp] Serving notebooks from local directory: C:\Users\Dir\_NITTTR', '[I 09:43:53.028 NotebookApp] The Jupyter Notebook is running at:', '[I 09:43:53.028 NotebookApp] http://localhost:8888/?token=86cc373d662c45467c27876c25e391245e473b584606a314', '[I 09:43:53.028 NotebookApp] or http://127.0.0.1:8888/?token=86cc373d662c45467c27876c25e391245e473b584606a314', '[I 09:43:53.028 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).', and '[C 09:43:53.119 NotebookApp]'. Below the log messages, there is a section titled 'To access the notebook, open this file in a browser:' followed by the file path 'file:///C:/Users/Dir\_NITTTR/AppData/Roaming/jupyter/runtime/nbserver-32396-open.html'. Below that, it says 'Or copy and paste one of these URLs:' followed by the two URLs: 'http://localhost:8888/?token=86cc373d662c45467c27876c25e391245e473b584606a314' and 'or http://127.0.0.1:8888/?token=86cc373d662c45467c27876c25e391245e473b584606a314'.

```
(base) C:\Users\Dir_NITTTR> jupyter notebook
[I 09:43:53.022 NotebookApp] JupyterLab extension loaded from C:\Users\Dir_NITTTR\Anaconda3\lib\site-packages\jupyterlab
[I 09:43:53.022 NotebookApp] JupyterLab application directory is C:\Users\Dir_NITTTR\Anaconda3\share\jupyter\lab
[I 09:43:53.027 NotebookApp] Serving notebooks from local directory: C:\Users\Dir_NITTTR
[I 09:43:53.028 NotebookApp] The Jupyter Notebook is running at:
[I 09:43:53.028 NotebookApp] http://localhost:8888/?token=86cc373d662c45467c27876c25e391245e473b584606a314
[I 09:43:53.028 NotebookApp] or http://127.0.0.1:8888/?token=86cc373d662c45467c27876c25e391245e473b584606a314
[I 09:43:53.028 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 09:43:53.119 NotebookApp]

To access the notebook, open this file in a browser:
  file:///C:/Users/Dir_NITTTR/AppData/Roaming/jupyter/runtime/nbserver-32396-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=86cc373d662c45467c27876c25e391245e473b584606a314
  or http://127.0.0.1:8888/?token=86cc373d662c45467c27876c25e391245e473b584606a314
```

Fig -14

c. Now we get the following home page: -

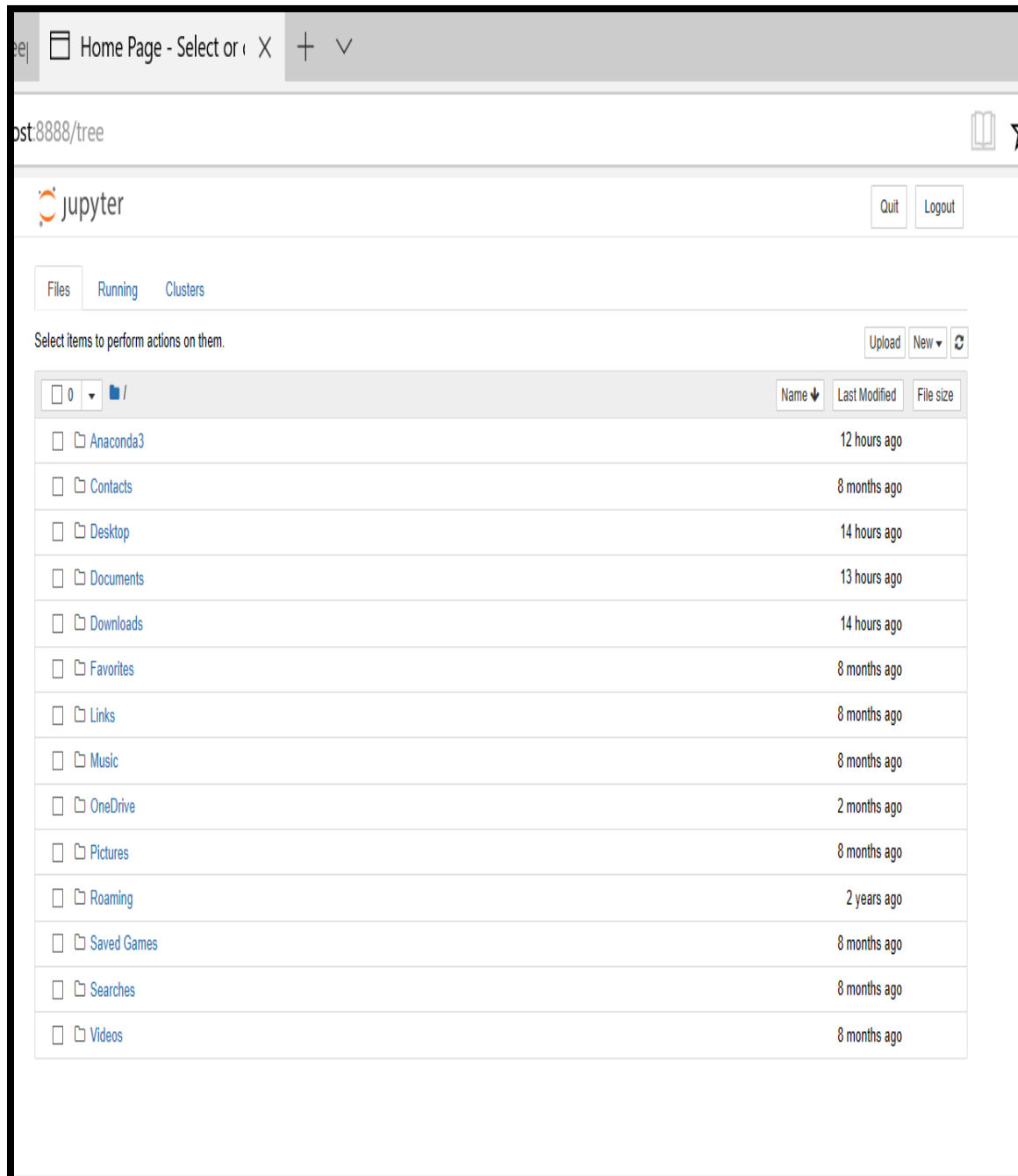


Fig-15

d. Click on “NEW” we get the following: -



Fig – 16

e. After clicking on new click on python 3 as shown:

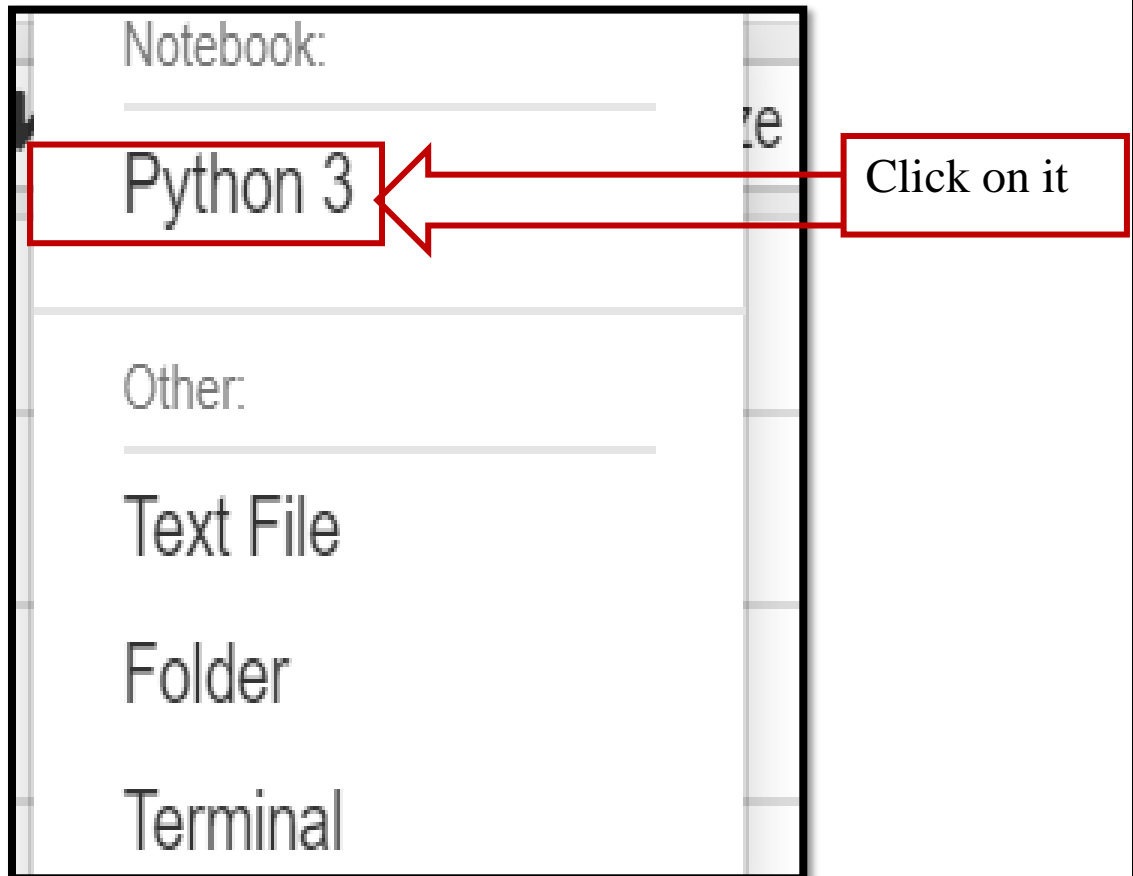


Fig – 17

f. Then we get the following:-

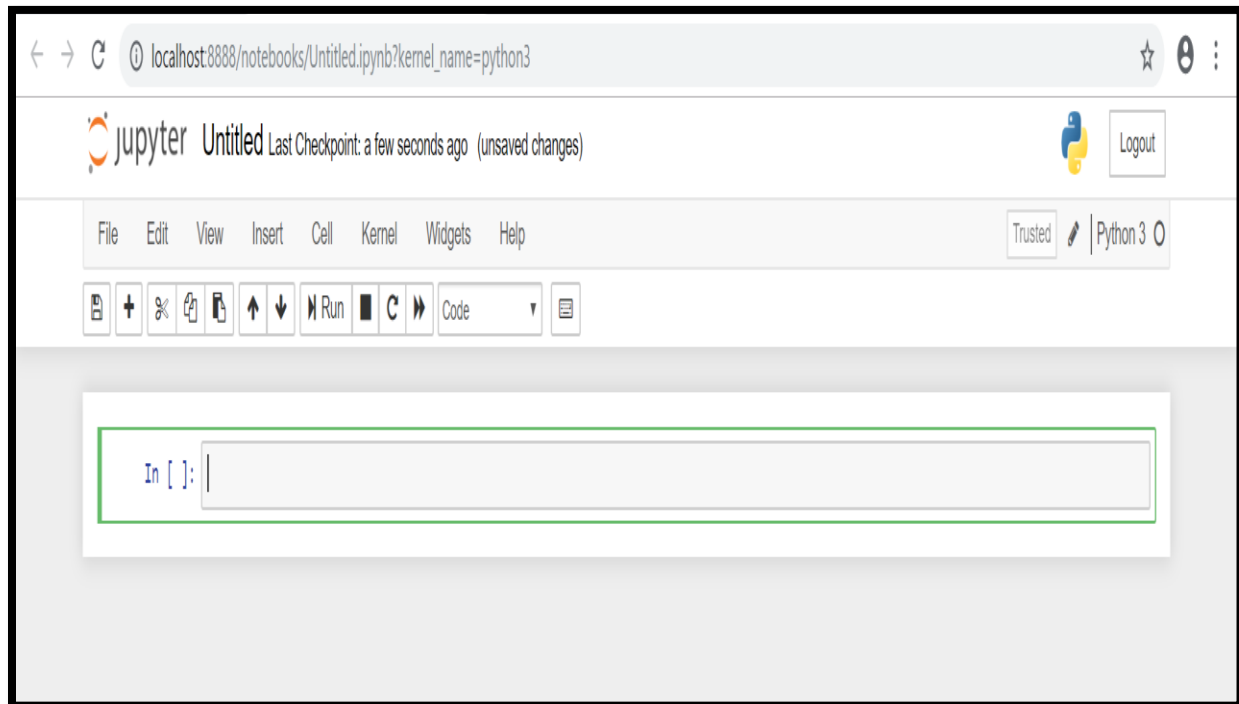


Fig-18

g. As shown in the following: -



Fig 19

h. To run the program, click on the following: -



Fig -20

2. Steps to rename Jupyter: -

First go to file on Jupyter Notebook as shown: -

1) Click there and we get the following figure.



Fig-21

2) Click on the Rename.

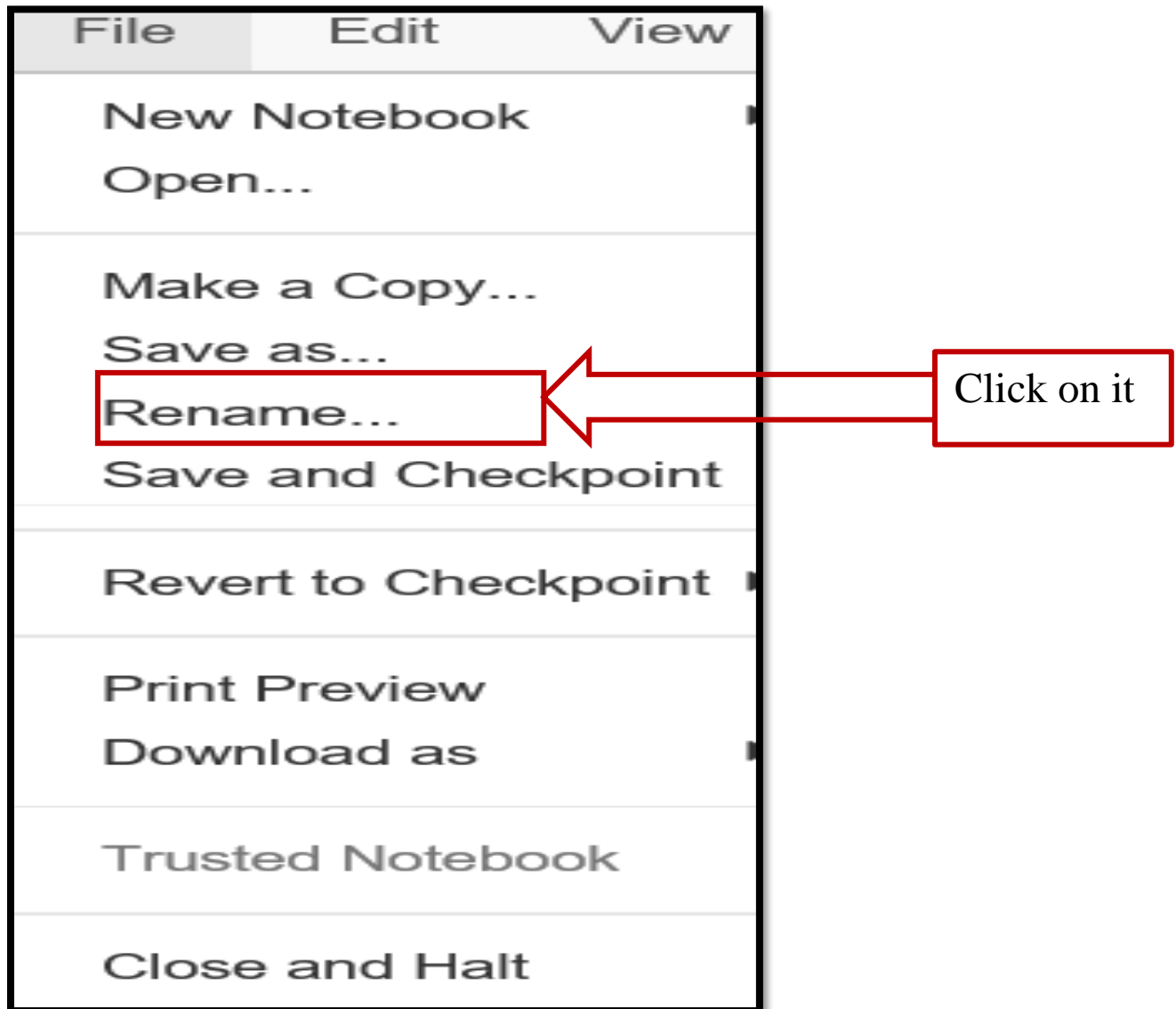


Fig-22

3) We get the following:-

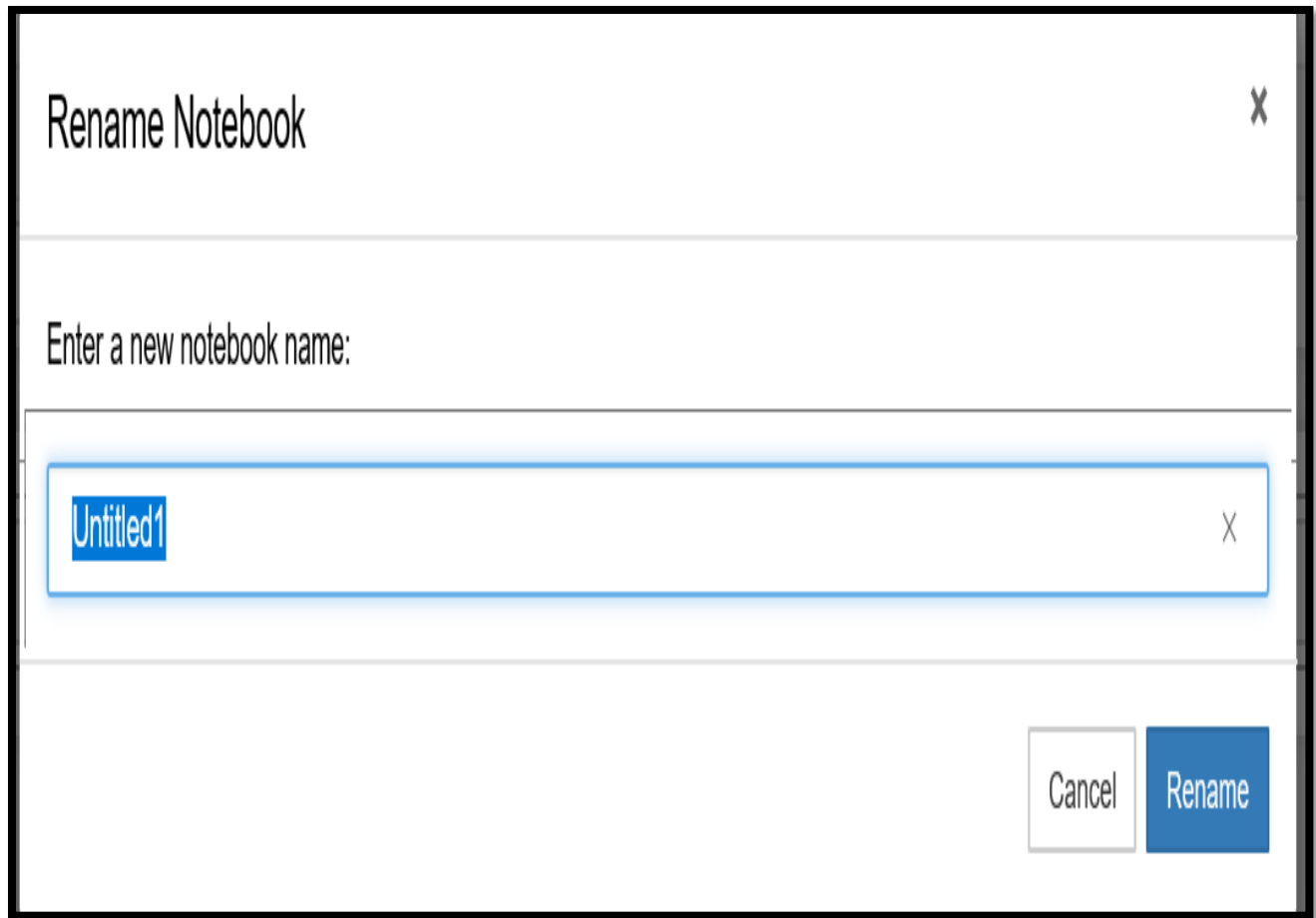


Fig-23

Now we can rename as we want to save.

# Basic On Python

## Introduction on PYTHON: -

- Python is one of the most popular programming language created by Guido van Rossum.
- It is a general-purpose interpreted, interactive, object-oriented, and high-level programming language.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.
- Characteristics of Python

Following are important characteristics of Python Programming –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

## Why Python?

- Python is one of the easiest language, which is readable and understandable.
- Here the codes written is nearer to the English language.
- There is no such restriction in the language, so it is highly popular among the developers.



## Basic:

a. For initializing variables in python: -

i. <variables name> = <value>

ii. For taking input from the user: -

<variable name> = input (“enter the name:”)

Examples “Hello Jupyter Programmers!!!”

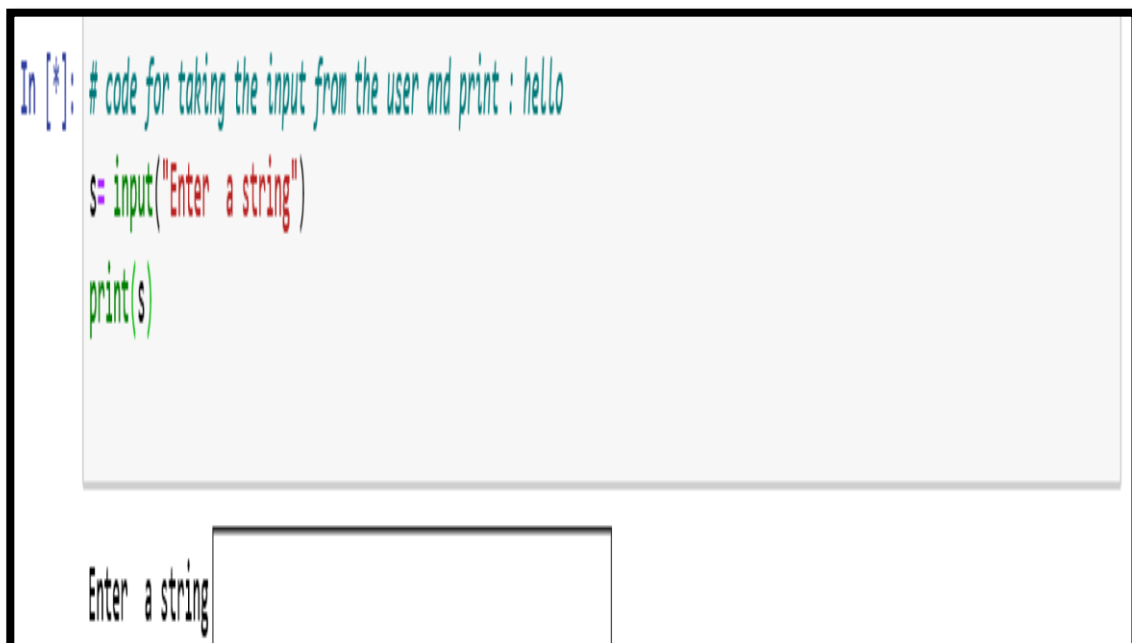


```
In [6]: print("Jupyter programmer!!")
```

Jupyter programmer!!

Fig – 24

The program for taking an input from the user and print “hello”



```
In [*]: # code for taking the input from the user and print : hello
s= input("Enter a string")
print(s)
```

Enter a string

Fig - 25

b. Blok of Indentation: -

In python indentation is used to define the loop and the control structure. Here the user has to pay the attention to the whitespaces.

Like other language to define the starting of the block in the function they use curly braces “{}” here in python it uses colons “:”.

Examples:

```
def symbol (): #user defined function and operation
    a = a + 1
    return a
print (a)
```

Comment line



c. Tables of key words in Python: -

true	False	none	and
Or	Asser t	Brea k	continu e
Class	Def	If	elif
Else	Del	Try	except
Raise	For	Whil e	pass
Import	Finall y	Fro m	as
Lambd a	Retur n	With	in
Yield	As	Is	global
Nonloc al			

Table -1

#### d. Decision making

##### i. if conditional statement: -

```
if condition:  
    # Statements to execute if  
    # condition is true
```

Fig - 26

##### ii. nested if: -

```
if (condition1):  
    # Executes when condition1 is true  
    if (condition2):  
        # Executes when condition2 is true  
    # if Block is end here  
# if Block is end here
```

Fig -27

### iii. if elif else ladder

```
if (condition):  
    statement  
elif (condition):  
    statement  
.  
.  
else:  
    statement
```

Fig – 28

### e. Loops in Python

#### 1. while loop:

```
while expression:  
    statement(s)
```

Fig -29

2. for loop:

```
for iterator_var in sequence:  
statements (s)
```

Fig -30

Functions in python: -

A function is an organized reusable code which performs and defines certain form of task.

Syntax: -

```
def functionname(parameters):
```

```
    “function expression task and operation”
```

```
return [expression]
```

# Python 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. 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. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science. This is the main reason why NumPy is much faster than lists. Also it is optimized to work with latest CPU architectures. Some python distribution already have NumPy installed like, Anaconda, and Spyder etc.

## Arrays in NumPy

Array in Numpy is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In Numpy, number of dimensions of the array is called rank of the array. A tuple of integers giving the size of the array along each dimension is known as shape of the array. An array class in Numpy is called as **ndarray**. Arrays in Numpy can be created by multiple ways, with various number of Ranks, defining the size of the Array. Arrays can also be created with the use of various data types such as lists, tuples, etc.

# Import NumPy

```
import numpy as np # importing numpy
```

Fig 31

Simple Array program :-

## create an array using numpy

```
In [1]: import numpy as np # importing numpy
arr = np.array([[1, 2, 3],
                [4, 5, 6]])
print("the rank", arr)
```

```
the rank [[1 2 3]
          [4 5 6]]
```

Fig- 32

# Pandas Tutorial

Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. It is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the Data Frame. Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables.

## Key Features of Pandas

1. Fast and efficient DataFrame object with default and customized indexing.
2. Tools for loading data into in-memory data objects from different file formats.
3. Data alignment and integrated handling of missing data.
4. Reshaping and pivoting of date sets.
5. Label-based slicing, indexing and sub-setting of large data sets.
6. Columns from a data structure can be deleted or inserted.
7. Group by data for aggregation and transformations.
8. High performance merging and joining of data.
9. Time Series functionality.

Note:-Anaconda Python package, Pandas will be installed by default.



We import pandas in anaconda as:-

---

```
In [ ]: import pandas as pd
```

---

Fig-33

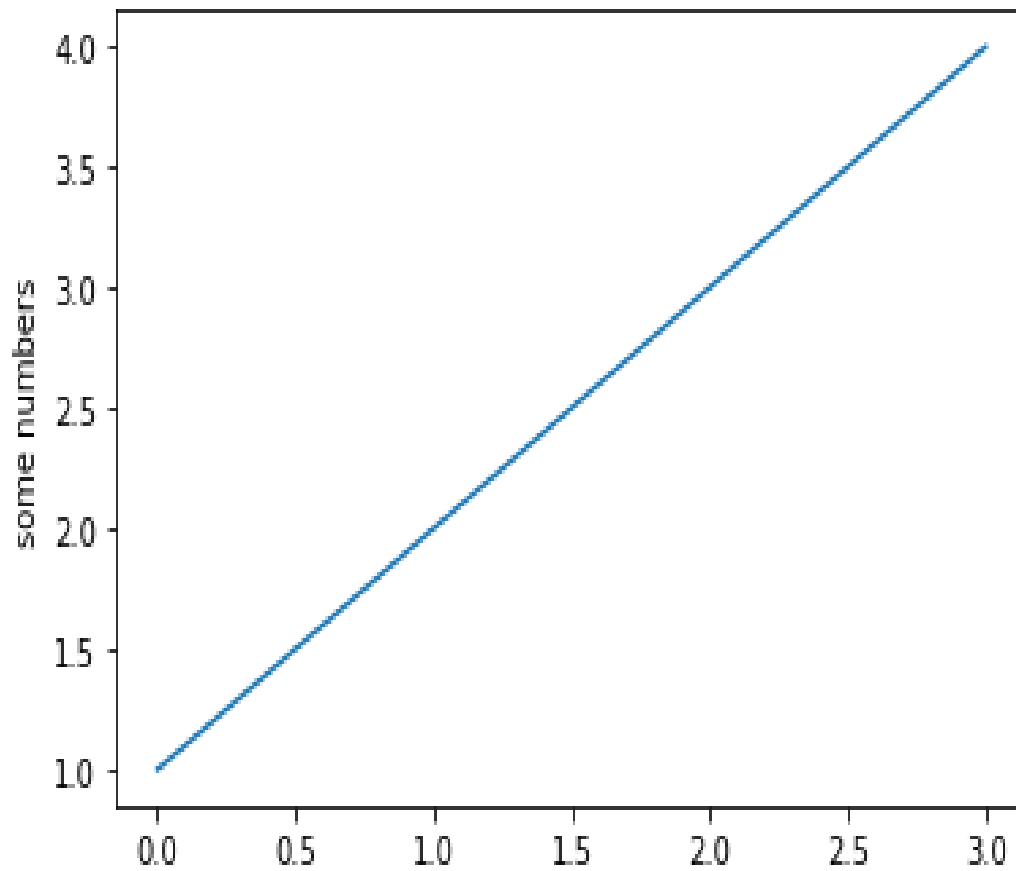
## Matplotlib

**Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged

[matplotlib.pyplot](#) is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. Matplotlib was originally written by John D. Hunter.

Example on matplotlib:-

```
: import matplotlib.pyplot as plt  
plt.plot([1, 2, 3, 4])  
plt.ylabel('some numbers')  
plt.show()
```



---

Fig-33

# BASIC CONSTRUCTION OF RECOMMENDER SYSTEM

Here we create a hybrid based filtering basic recommender system using python and its library. Here we collect the dataset and normalized the data.

## STEP 1:- Collection of The Dataset.

Download the data set from the following link as shown:-

<https://www.kaggle.com/tmdb/tmdb-movie-metadata>

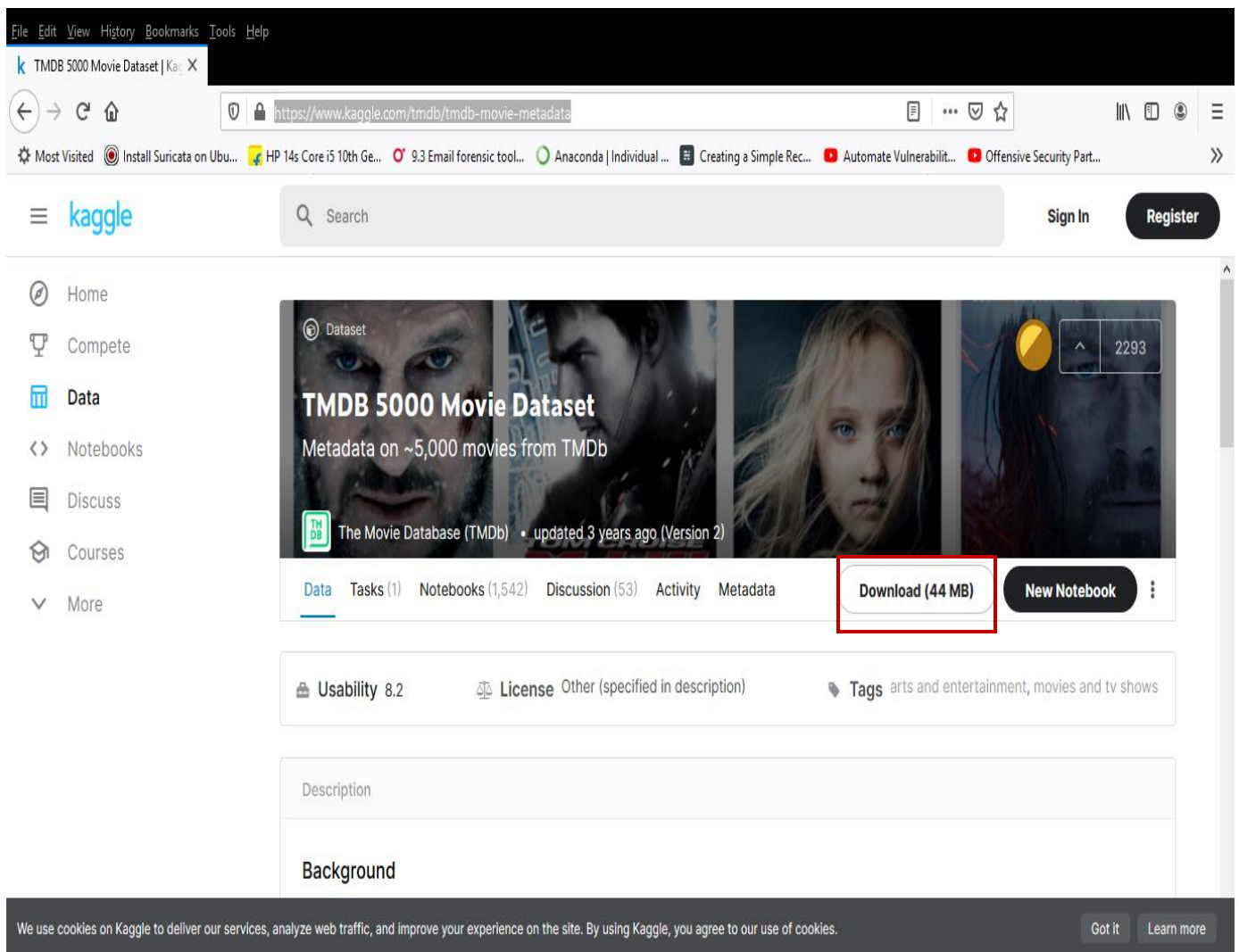


Fig 34

Now we can see that the dataset has been downloaded as csv format.

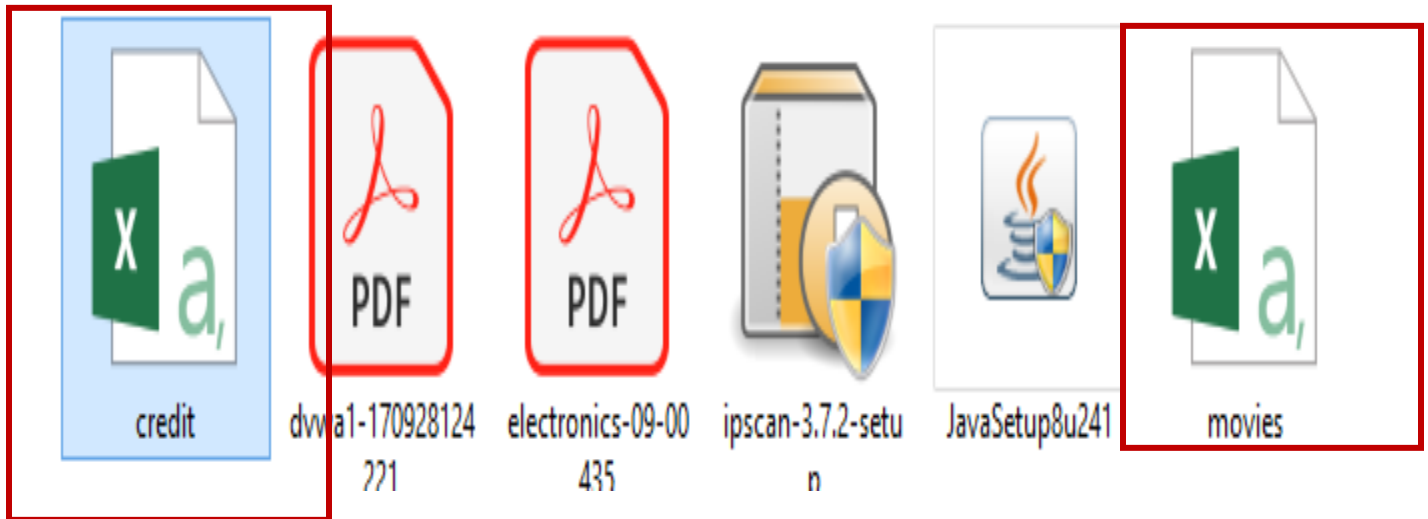


Fig- 35

Step 2:- Now Open Anaconda Command Prompt For Opening Jupyter Notebook.

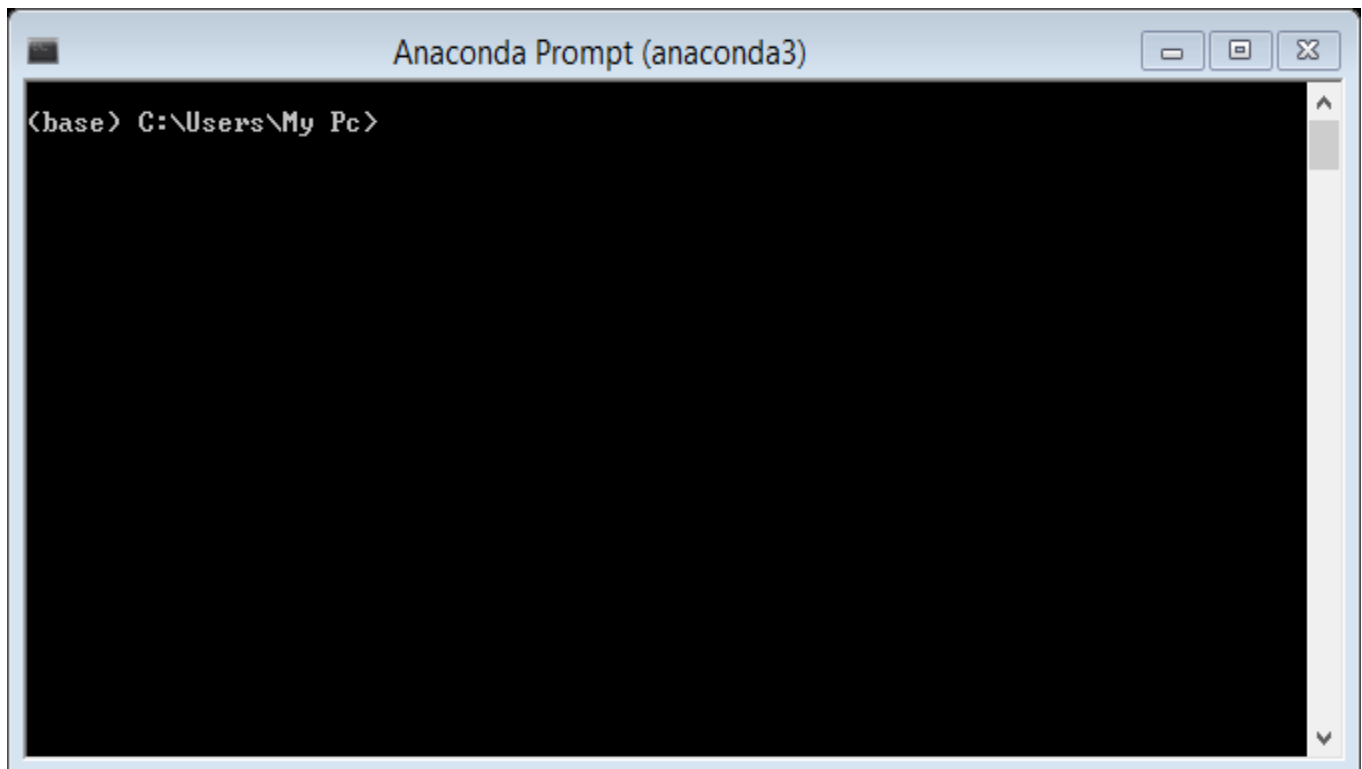


Fig-36

Now write the following command to open jupyter notebook.  
“jupyter notebook” and press enter.

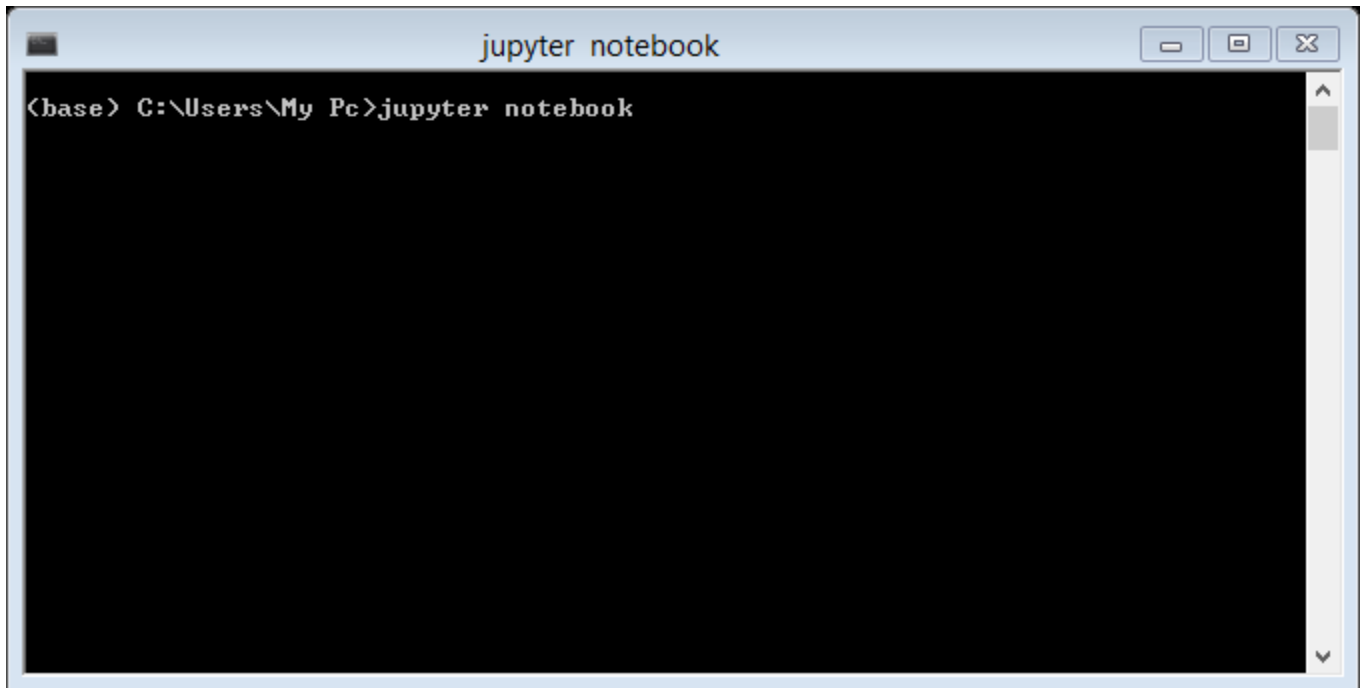


Fig-37

Now we can see that notebook has started:-

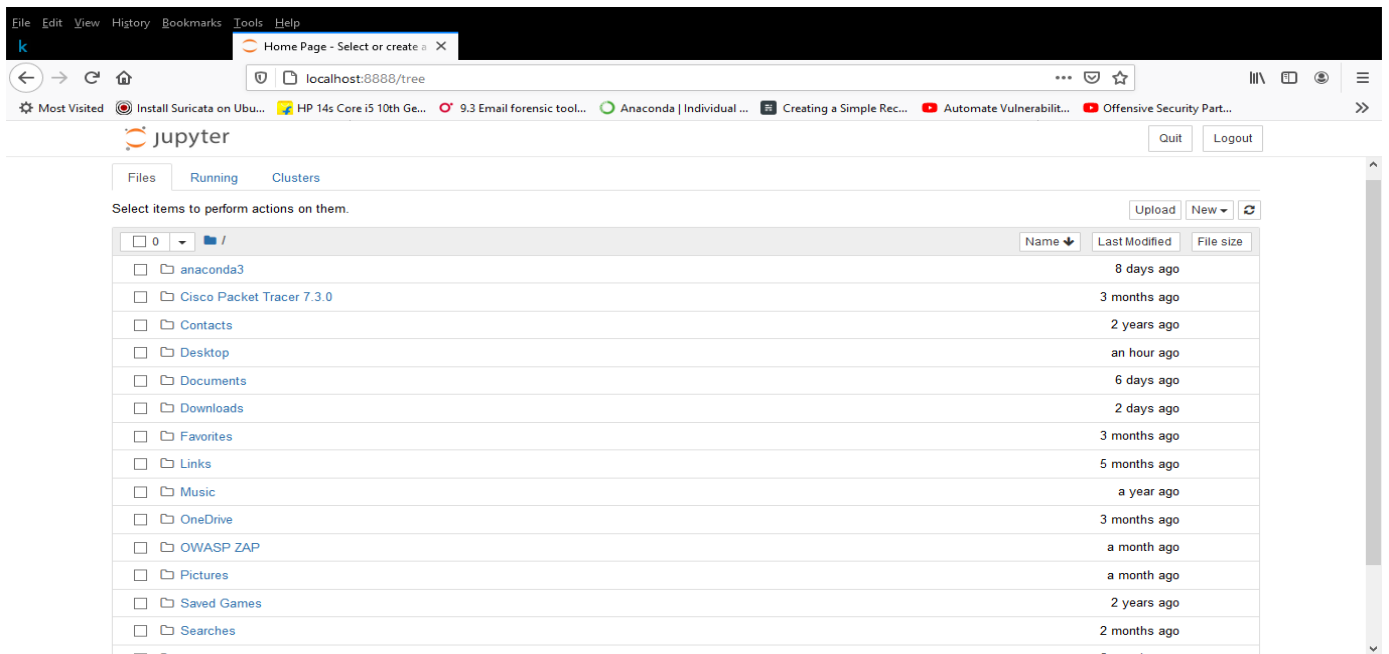


Fig-38

### Step 3:- start python program.

1. Click on "NEW".

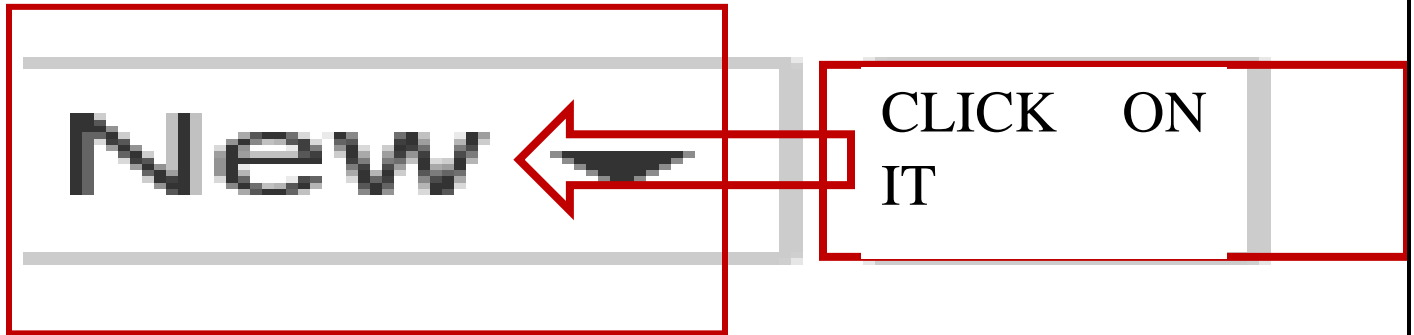


Fig-39

2. After clicking on new click on python 3 as shown:

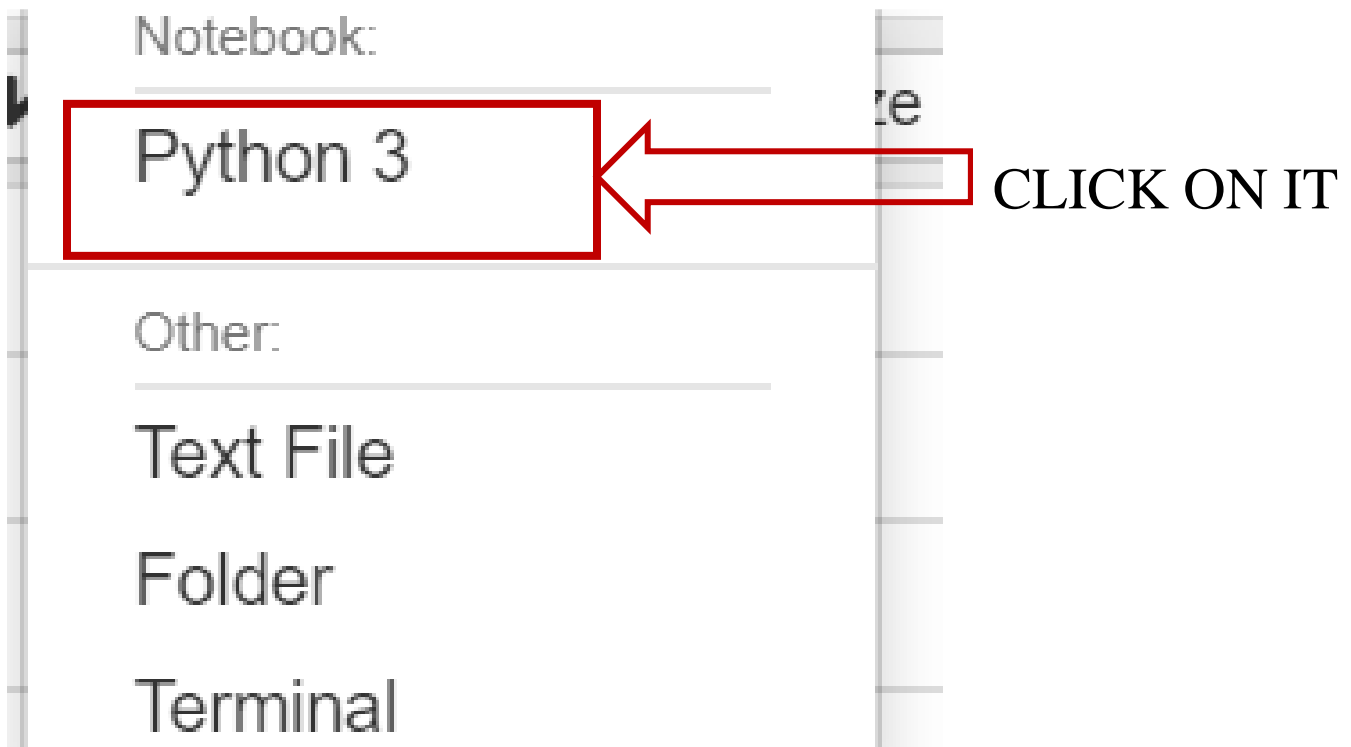


Fig- 40

3. Then we get the following:-

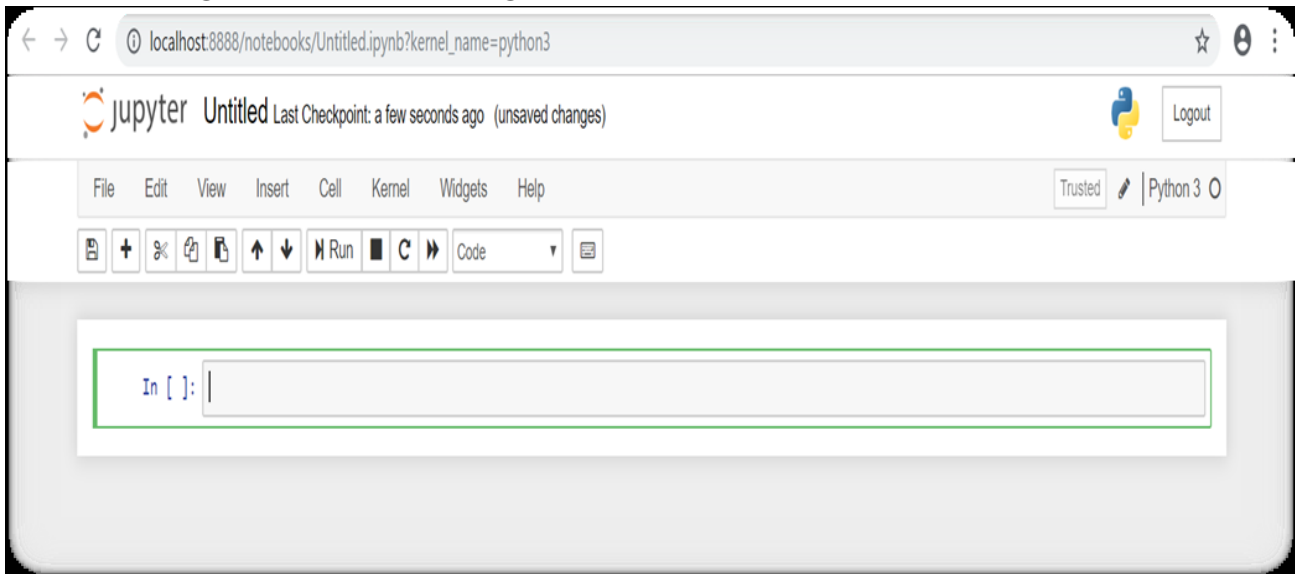


Fig-41

## STEP 4: IMPORT PANDAS AND NUMPY.

As we import the NumPy and Pandas we also link the dataset.



Fig- 42

# Step 5: Output of Movie Dataset and Credit Dataset.

```
In [6]: credit.head()
```

```
Out[6]:
```

	movie_id	title	cast	crew
0	19995	Avatar	[[{"cast_id": 242, "character": "Jake Sully", "...	[[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	[[{"cast_id": 4, "character": "Captain Jack Spa...	[[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	[[{"cast_id": 1, "character": "James Bond", "cr...	[[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	[[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	[[{"cast_id": 5, "character": "John Carter", "c...	[[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

Fig-43 (credit set)

```
In [8]: movie_df.head()
```

```
Out[8]:
```

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies
0	237000000	[[{"id": 28, "name": "Action"}, {"id": 12, "name": "Sci-Fi"}]]	http://www.avatarmovie.com/	19995	[[{"id": 1463, "name": "culture clash"}, {"id": 1464, "name": "marine"}]]	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[[{"name": "Ingenious Film Partners", "id": 289...}]]
1	300000000	[[{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}]]	http://disney.go.com/disneypictures/pirates/	285	[[{"id": 270, "name": "ocean"}, {"id": 726, "name": "pirates"}]]	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[[{"name": "Walt Disney Pictures", "id": 2}, {"name": "Disneytoon Studios", "id": 3}]]
2	245000000	[[{"id": 28, "name": "Action"}, {"id": 12, "name": "Sci-Fi"}]]	http://www.sonypictures.com/movies/spectre/	206647	[[{"id": 470, "name": "spy"}, {"id": 818, "name": "international relations"}]]	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	[[{"name": "Columbia Pictures", "id": 5}, {"name": "United Artists", "id": 6}]]
3	250000000	[[{"id": 28, "name": "Action"}, {"id": 80, "name": "Drama"}]]	http://www.thedarkknightrises.com/	49026	[[{"id": 849, "name": "dc comics"}, {"id": 853, "name": "superhero"}]]	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	[[{"name": "Legendary Pictures", "id": 923}, {"name": "Warner Bros. Entertainment", "id": 924}]]



Fig-44(movie dataset)

## Step 6: Shape of The Dataset (Describe Nos of Columns And Rows)

```
In [9]: print("credits of the movie:", credit.shape)
        print("dataframe of the movie:", movie_df.shape)

credits of the movie: (4803, 4)
dataframe of the movie: (4803, 20)
```

Fig- 45

## Step 7: Merging of Both the Table.

As the both table contains same contents so we will merge them.

```
In [10]: credit_rename=credit.rename(index=str, columns={"movie_id": "id"})
        movie_df_merge=movie_df.merge(credit_rename, on='id')
        movie_df_merge.head()
```

Fig-46(input)

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies
0	237000000	[[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]]	http://www.avatarmovie.com/	19995	[[{"id": 1463, "name": "culture clash"}, {"id": 1464, "name": "culture clash"}]]	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[[{"name": "Ingenious Film Partners", "id": 289}]]
1	300000000	[[{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Action"}]]	http://disney.go.com/disneypictures/pirates/	285	[[{"id": 270, "name": "ocean"}, {"id": 726, "name": "ocean"}]]	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[[{"name": "Walt Disney Pictures", "id": 2}, {"name": "Disney", "id": 1}]]
2	245000000	[[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]]	http://www.sonypictures.com/movies/spectre/	206647	[[{"id": 470, "name": "spy"}, {"id": 818, "name": "spy"}]]	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	[[{"name": "Columbia Pictures", "id": 5}, {"name": "Sony Pictures Entertainment", "id": 1}]]
3	250000000	[[{"id": 28, "name": "Action"}, {"id": 80, "name": "Action"}]]	http://www.thedarkknightrises.com/	49026	[[{"id": 849, "name": "dc comics"}, {"id": 853, "name": "dc comics"}]]	en	The Dark Knight Rises	Following the death of District Attorney Harvey Dent...	112.312950	[[{"name": "Legendary Pictures", "id": 923}, {"name": "Warner Bros. Entertainment", "id": 1}]]
4	260000000	[[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]]	http://movies.disney.com/john-carter	49529	[[{"id": 818, "name": "based on novel"}, {"id": 819, "name": "based on novel"}]]	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	[[{"name": "Walt Disney Pictures", "id": 2}, {"name": "Disney", "id": 1}]]

Fig 47(output)

## Step 8: We Will Remove Unwanted Data From The Merged Table.

```
In [11]: movie_df_datas= movie_df_merge.drop(columns = ['budget', 'genres', 'homepage', 'keywords', 'original_language',
                                                    'production_companies', 'runtime', 'overview',
                                                    'title_y', 'revenue', 'cast', 'crew', 'production_countries'])

movie_df_datas.head()
```

Out[11]:

	id	original_title	popularity	release_date	vote_average	vote_count
0	19995	Avatar	150.437577	2009-12-10	7.2	11800
1	285	Pirates of the Caribbean: At World's End	139.082615	2007-05-19	6.9	4500
2	206647	Spectre	107.376788	2015-10-26	6.3	4466
3	49026	The Dark Knight Rises	112.312950	2012-07-16	7.6	9106
4	49529	John Carter	43.926995	2012-03-07	6.1	2124

Fig- 48

## Step 9: Check The Relevant Data Present or Not.

```
In [12]: movie_df_datas.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4803 entries, 0 to 4802
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    4803 non-null   int64
1   original_title        4803 non-null   object
2   popularity            4803 non-null   float64
3   release_date          4802 non-null   object
4   vote_average          4803 non-null   float64
5   vote_count            4803 non-null   int64
dtypes: float64(2), int64(2), object(2)
memory usage: 262.7+ KB
```

Fig -49

## Step10: Weighted Hybrid Based Filtering Construction.

### # weighted average movie rating

W= weighted rating

m= minimum required vote to be listed in top 250 (currently 300)

C= mean vote across the data (currently 6.9)

v= nos of votes for the movie=(votes)

R = average for the movie as a number from 0 to 10 (mean)=(rating)

$W = Rv + Cm / v + m$

[http://trailerpark.weebly.com/imdb-rating.html?source=post\\_page](http://trailerpark.weebly.com/imdb-rating.html?source=post_page)

Fig-50 (formula)

## Step 11: Calculate Mean of Voting Average And Percentage Of Vote Count.

In [13]:

```
m=movie_df_datos['vote_count'].quantile(0.65)|
v=movie_df_datos['vote_count']
R=movie_df_datos['vote_average']
C=movie_df_datos['vote_average'].mean()
```

Fig -51

## Step 12: Calculate Weight.

```
In [14]: movie_df_datas['weight']=( (R*v)+(C*m) ) / (v+m)

In [15]: movie_df_datas['weight'].head()

Out[15]: 0      7.158147
         1      6.824593
         2      6.280466
         3      7.526998
         4      6.098598
         Name: weight, dtype: float64
```

Fig -52

## Step 13: Output of The Movie Dataset.

```
In [17]: movie_df_datas.head(10)

Out[17]:
```

	id	original_title	popularity	release_date	vote_average	vote_count	weight
0	19995	Avatar	150.437577	2009-12-10	7.2	11800	7.158147
1	285	Pirates of the Caribbean: At World's End	139.082615	2007-05-19	6.9	4500	6.824593
2	206647	Spectre	107.376788	2015-10-26	6.3	4466	6.280466
3	49026	The Dark Knight Rises	112.312950	2012-07-16	7.6	9106	7.526998
4	49529	John Carter	43.926995	2012-03-07	6.1	2124	6.098598
5	559	Spider-Man 3	115.699814	2007-05-01	5.9	3576	5.922042
6	38757	Tangled	48.681969	2010-11-24	7.4	3330	7.240267
7	99861	Avengers: Age of Ultron	134.279229	2015-04-22	7.3	6767	7.222605
8	767	Harry Potter and the Half-Blood Prince	98.885637	2009-07-07	7.4	5293	7.294738
9	209112	Batman v Superman: Dawn of Justice	155.790452	2016-03-23	5.7	7004	5.724332

Fig -53

## Step 13: Arrange in Ascending Order.

```
In [19]: movies_rank = movie_df_datas.sort_values('weight', ascending= True )
movies_rank[['id', 'original_title', 'popularity', 'release_date', 'vote_average', 'vote_count', 'weight']].head(10)
```

Out[19]:

	id	original_title	popularity	release_date	vote_average	vote_count	weight
1652	14164	Dragonball Evolution	21.677732	2009-04-01	2.9	462	4.498328
210	415	Batman & Robin	50.073575	1997-06-20	4.2	1418	4.665977
242	166424	Fantastic Four	38.126095	2015-08-05	4.4	2278	4.685990
3746	241251	The Boy Next Door	24.161735	2015-01-23	4.1	1022	4.721405
303	314	Catwoman	32.271938	2004-07-22	4.2	808	4.889564
2122	9760	Epic Movie	6.064638	2007-01-25	3.2	326	4.897635
3843	205321	Sharknado	20.466433	2013-07-11	3.8	476	4.930590
499	71880	Jack and Jill	22.132418	2011-11-11	4.1	604	4.964774
480	5491	Battlefield Earth	7.891470	2000-05-10	3.0	255	4.994436
2320	218043	Left Behind	33.452469	2014-10-03	3.7	392	4.995795

Fig- 54

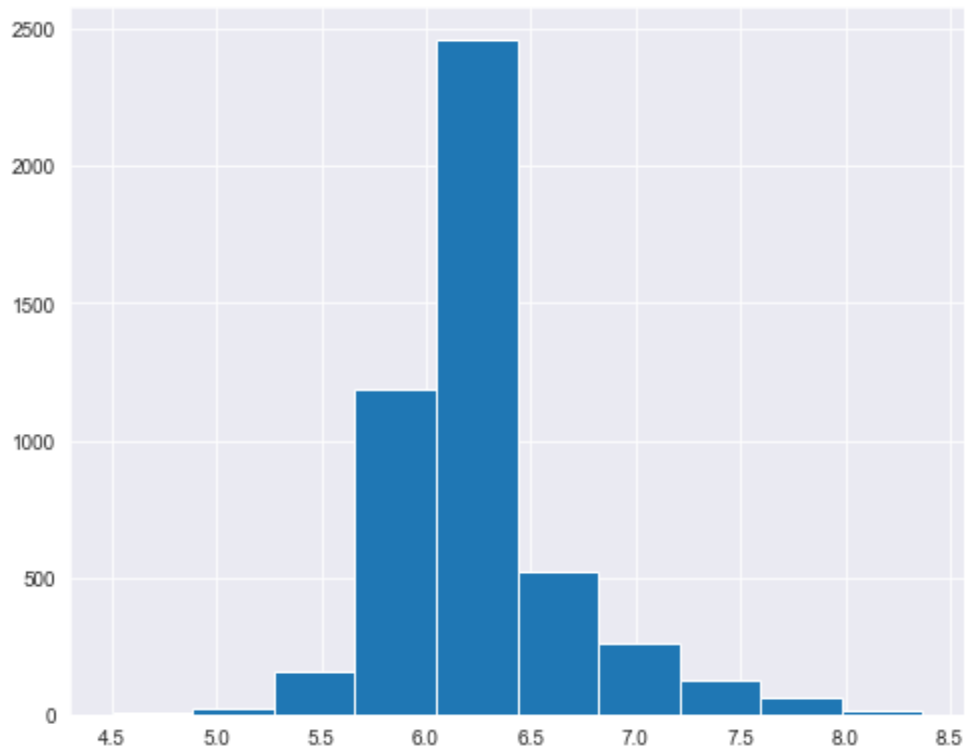
## Step 14: Use of Matplotlib to Get Graphical Data.

```
In [20]: import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('dark')
%matplotlib inline

plt.figure(figsize=(8,6))
plt.rcParams['patch.force_edgecolor'] = True
movies_rank['weight'].hist()
```

Fig -55

Out[20]: <matplotlib.axes.\_subplots.AxesSubplot at 0xc1bd546088>



Graph -1

## Step 15: Using Sklearn and MaxMinscaler We will Normalized Data to Reduce the Gap Between Them.

---

```
In [30]: from sklearn.preprocessing import MinMaxScaler
scaling= MinMaxScaler()
movie_df_scaling = scaling.fit_transform(movie_df_datas[['id','popularity', 'weight']])
movie_df_normalized = pd.DataFrame(movie_df_scaling, columns =['id','popularity', 'weight'])
movie_df_normalized.head(10)
```

Out[30]:

	id	popularity	weight
0	0.043505	0.171815	0.686763
1	0.000609	0.158846	0.600640
2	0.449727	0.122635	0.460147
3	0.106687	0.128272	0.782000
4	0.107782	0.050169	0.413188
5	0.001206	0.132141	0.367602
6	0.084338	0.055600	0.707966
7	0.217323	0.153360	0.703406
8	0.001658	0.112937	0.722031
9	0.455092	0.177928	0.316553

---

Fig- 56



# Benefits in Recommender System

1. Benefits of recommender systems are:
2. Revenue—past years, many researchers have studied and generate many algorithms to learn increasing rate for an online customer like Amazon site. Also, These algorithms study the difference between shopping online sites with others using recommender systems for items to increase revenue by increasing the number of sales.
3. Client Satisfaction— many times customers tend to expect to see near similar product recommendation from their last browsing search on the site. Mainly because they believe they will get more serious chances for better products. When they leave the situation and get back afterward; it would assist if their browsing data from the previous shopping or viewing product list. This could further facilitate and guide their e-Commerce activities, similar two experienced assistants. This case of client satisfaction contributes to client retention.
4. Personalization—we often get recommendations from our friends. They recognize what we like better than anyone else. This is the only reason they are adept at recommending things and is what recommendation systems try to model. You can utilize the data collected indirectly to improve your website's overall services and assure that they are suitable according to a user's preference.
5. Discovery—people need to be recommended items they would like or prefer, and when they find a web page for shopping or movie, songs, etc. meet their hopes they bound to visit this site again.
6. Provide Reports—is an integral piece of a personalization scheme. Making the client accurate and up to the minute, reporting allows him to make strong conclusions about his site and the management of a movement. Founded on these reports clients can get offers for slow-moving products in order to make a drive in sales.

# CONCLUSION

A recommender system has been a hot topic for a long time. They are simple algorithms which aim to provide the most relevant and accurate items to the user by filtering useful stuff from of a huge pool of information base. Recommender engines discovers data patterns in the data set by learning consumer's choices and produces the outcomes that co-relates to their needs and interests.

# Assignment Questions

1. What is data science?
2. What is data mining?
3. Define machine learning?
4. Define information filtering?
5. Types of information filtering?
6. Define recommender system.
7. Why recommender system?
8. Benefits of recommender system.
9. Difference between data mining and data filtering?
10. Give differences between data science and machine learning?
11. What are the problems faced in recommender system?
12. Different way of creating recommender system?
13. Different algorithm used in creating recommender system?
14. Give a summary on few algorithm?
15. What are the security problem faced by recommender system?
16. State different types of learning?
17. State different types of feedback?
18. Why matplotlib is used?
19. Create a graphical representation showing annual growth in blockbuster movies using python.
20. Create a collaborative model using python using any data set.
21. Using any language of your choice create a basic recommender system.