A Project report

on

Recommendation System using Machine Learning

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

S. Ravali	**(174G1A0564)**
P. Subhashini	**(174G1A0594)**
V. Rohith Sai	**(174G1A0567)**
S. Thahaseen Ruqhiya	**(174G1A05A0)**

Under the Guidance of

Dr. B. Hari Chandana, M.Tech., Ph.D
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPURAMU
(Affiliated to JNTUA, Approved by AICTE, New Delhi, Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE & CSE))
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu – 515701

2020-2021

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPURAMU
(Affiliated to JNTUA, Approved by AICTE, New Delhi, Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE & CSE))
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu – 515701

Certificate

This is to certify that the project report entitled **Recommendation Systems using Machine Learning** is the bonafide work carried out by **S. Ravali** bearing Roll Number **174G1A0564, P. Subhashini** bearing Roll Number **174G1A0594, V. Rohith Sai** bearing Roll Number **174G1A0567, S. Thahaseen Ruqhiya** bearing Roll Number **174G1A05A0** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2020-2021.

Signature of the Guide

Dr. B.Hari Chandana, M.Tech., Ph.D
Associate Professor

Head of the Department

Dr. G.K.V.Narasimha Reddy, Ph.D
Professor

Date:
Place: Rotarypuram

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Dr. B. Hari Chandana, Associate Professor**, **Computer Science & Engineering,** who has guided us a lot and encouraged us in every step of the project work. We thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep felt gratitude to **Dr. P. Chitralingappa, Associate Professor and Mrs. M. Soumya,** project coordinator valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We are very much thankful to **Dr. G. K. V. Narasimha Reddy, Professor and Head of Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Balakrishna, Principal, Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time. We also express our sincere thanks to the Management for providing excellent facilities.

Finally, we wish to convey our gratitude to our families who fostered all requirements and facilities that we need.

Project Associates

DECLARATION

We, Ms. S. Ravali with reg no.: 174G1A0564, Ms. P. Subhashini with reg no.: 174G1A0594, Mr. V. Rohith Sai with reg no.: 174G1A0567, Ms. S. Thahaseen Ruqhiya with reg no.: 174G1A05A0 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that the dissertation entitled "RECOMMENDATION SYSTEM USING MACHINE LEARNING" embodies the report of our project work carried out by us during IV year Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING, under the supervision of Dr. B. Hari Chandana, Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project report have not been submitted to any other University of Institute for the award of any Degree or Diploma.

S. Ravali	Reg no.: 174G1A0564
P. Subhashini	Reg no.: 174G1A0594
V. Rohith Sai	Reg no.: 174G1A0567
S. Thahaseen Ruqhiya	Reg no.: 174G1A05A0

CONTENTS

List of Figures

List of Screens

LIST OF ABBREVIATIONS

QOS	Quality of service
NLTK	Natural Language Toolkit
UML	Unified Modeling Language
RTM	Requirements Traceability Matrix
KNN	K-Nearest Neighbor
UBCF	User Based Collaborative Filtering
PBCF	Product-Based Collaborative Filtering
IBCF	Item-Based Collaborative Filtering
CF	Collaborative Filtering
Numpy	Numerical Python
CSV	Comma Separated Value
SRS	Software Requirement Specification

ABSTRACT

Consumers currently enjoy a surplus of goods (books, videos, music, or other items) available to purchase. While this surplus often allows a consumer to find a product tailored to their preferences or needs, the volume of items available may require considerable time or effort on the part of the user to find the most relevant item. Recommender system creates similarities between the user and items and exploits the similarity between user/item. Recommendation systems have become a common part of many online business that supply users books, videos, music, or other items to consumers. These systems attempt to provide assistance to consumers in finding the items that fit their preferences.

This report presents an overview of recommendation systems. The classical methods for collaborative recommendation systems are reviewed. Product based and user-based filtering are implemented for the same dataset and the accuracy is calculated.

CHAPTER-1

INTRODUCTION

1.1 Background

The goal of a recommender system is to predict or recommend for every user, the items (video, audio, text, articles, ...) that would hold the greatest interest or satisfaction to that particular user. The standard recommendation systems in concerned with a set of users, U, and a set of items, I. Let there be a utility function, s, generally approximated by the set of ratings, R, that measures the usefulness, interest, or relevance of item i to user u. Initially, s is only defined with the elements in R.

The idea of a recommendation system was first proposed in the early 1990s. Its goal was to help Usenet users find interesting and useful content on the infant internet. The earliest approaches would now be classified as collaborative filtering, they sought to find similar users to the questor and utilize the data available to build personal filters. At the same time, the internet business bubble was just beginning to expand and recommendations systems were caught up with it. This increased commercialization of recommender systems drove improvement in a number of areas. First, recommender systems now needed to provide value in addition to the accuracy they were already demonstrating. Second, the size of the datasets being used had increased exponentially in the transition from a research to a commercial environment. Additionally, long delays in computation were no longer acceptable when being used in a rapidly changing online marketplace. Finally, marketing professionals were more interested in lists of items that would be most relevant to a user driving a shift in how these systems provide results.

1.1.1 Related works

The various features of machine learning, different models about it and emphasize how user recommendations system are related to the concept of machine learning.

Concept of Artificial Intelligence

Artificial intelligence is an important concept in the field of sciences in the world today. It involves using machines to develop a concept of intelligence in them which is more like human thinking. So, the way humans think, the machines also are taught various things which become a reason for them to evolve. There are various fields in which artificial intelligence used for the benefit of humanity.

Machine Learning in Artificial Intelligence

In the area of computer sciences, the term artificial intelligence is widely used and it connected deeply to concept of machine learning. Without a doubt, the concept of machine learning is related to the capability of machines to learn like humans. Strangely, this concept implies that the machine can learn things like a human mind. So, if it gives out recommendations about a user's experience, it might learn the next time about the preferences. of these users. It is based on the concept of self-learning about machines that they can learn and adjust to different kind of information received by them.

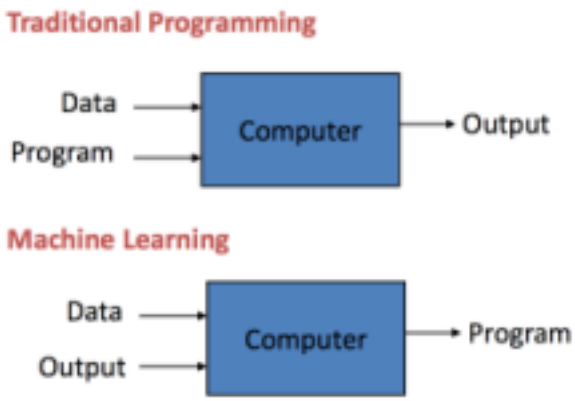


Figure 1.1. Machine Learning Vs Traditional Programming

There is a need to understand the working of machines that are capable of learning from experience. However, there are some differences that exist between machine learning and artificial intelligence. Let us review how these two things are different from each other on the basis of various factors. Artificial intelligence can be understood as the concept where the concept of intelligence is greatly encouraged. However, in machine learning, the people might be interested in the learning of a new

skill. Therefore, one can see that both of these concepts are useful in their own respects.

There is a slight difference between the goals for each concept in which one is about machine learning and other is about artificial intelligence. The feature of artificial intelligence is more like decision-making based and machine learning cannot deicide things on its own. It would not be wrong to say that the area of artificial intelligence serves an important role in wisdom while machine learning mostly concerns knowledge. In the area of artificial intelligence, one can find it working on the pattern of human mind. However, machine learning system operates on the basis of algorithms which perform art of self-learning.

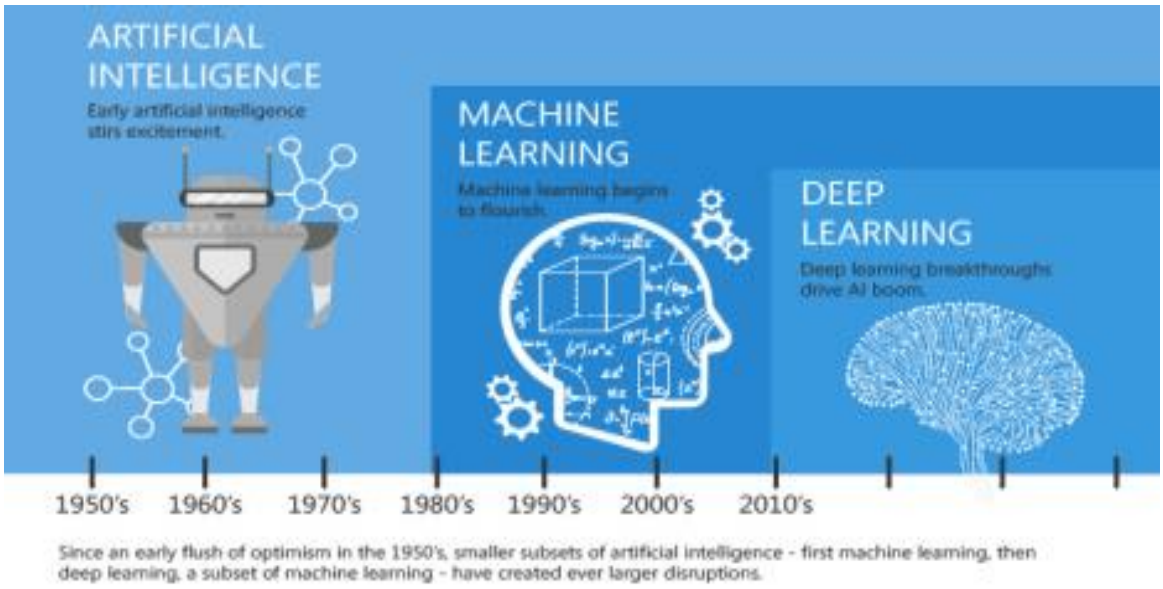


Figure 1.2. Artificial intelligence and machine learning

Machine Learning Types

Machine learning is a recent development in the field of artificial intelligence which can be further subdivided into various types. The three most important types of machine learning have been given below, namely supervised machine learning, unsupervised machine learning and reinforcement.

Supervised Machine Learning

The simpler form of machine learning consists of using supervised paradigm of machine learning. It is based on the assumption of using flash card that help people to

learn thing faster and better. However, there are certain form of algorithms which operate on the principle similar to flash cards.

Figure 1.3. Types of machine learning

Unsupervised Machine Learning

In unsupervised machine learning, labels are not used or identified as is the case with supervised machine learning. In this form of machine learning, there is lot of data available and direction is not given to the machine. Instead, the machine automatically learns to organize the data which is incorporated within it. The use of unlabeled in made so much here that the data becomes difficult to be managed. However, it helps the data systems to become organized even if zero supervision is given in this regard. There are some features in this form of learning including recommendation systems and group user logs.

Reinforcement Machine Learning

Reinforcement is a basic concept in the science of behaviourism or experimental psychology. It is used to explain the incentive-based system where learning is initiated on the basis of reinforcement. There is classical example of a dog in an experiment conducted by a scientist by the name of Pavlov. He stressed on the need that learning can be performed by means of repeated stimuli. So, he concluded from this research that humans and other animals have the ability to learn things. Therefore, they can be reinforced to perform things that are quite difficult for them to learn otherwise. In this case, reinforcement learning framework is the one which used this same method of reinforcement learning.

Significance of Machine Learning

There are various fields of life where machine learning is being utilized in order to perform various functions. It is used for the field of finance in which trading is being done for algorithm and credit scoring. The area of image processing is especially important in this regard where the process of object recognition is performed by using such models. For the area of computer vision, machine learning is quite important also where the task of face recognition as well as object recognition is done.

1.1.2 Recommender Systems

Recommendations systems can be categorized in a wide range of types. There are three basic data models, multiple types of input data, and two basic output formats used in all recommendation systems. The most successful systems seek to leverage all model types and harness all available data. The three types of computation models are: (i) collaborative/social-based filters, which use ratings from thousands of users and statistical analyses to find correlations among users and/or items to create predictions; (ii) content-based filters, which use data from description and/or analysis of their item set and the user's ratings to create predictions; and (iii) knowledge-based methods, which use ratings and the action history of the userbase to learn rules that describe desired items, then uses those rules to create predictions based on a user's explicitly stated preferences. Among the several types of system input are (i) text/video/audio, the actual content of the item or a description/analysis of the content; (ii) views/actions, can be binary in nature or numeric to capture repeated visits to, selection of, or sales of items; and (iii) ratings, generally an Likert scale with a numeric representation for expressing preferences (e.g., 1 = Hated It, 2 = Didn't Like It, 3 = Liked It, 4 = Really Liked It, 5 = Loved It).

The two types of system output are ratings and top-n lists. The first is a prediction of the rating given a specific user and item. The second looks at the predicted ratings for all unrated items to create a list of the n items that a user is most likely to enjoy. Below is a very simple illustration of how Recommender systems work in the context of an e-commerce site.

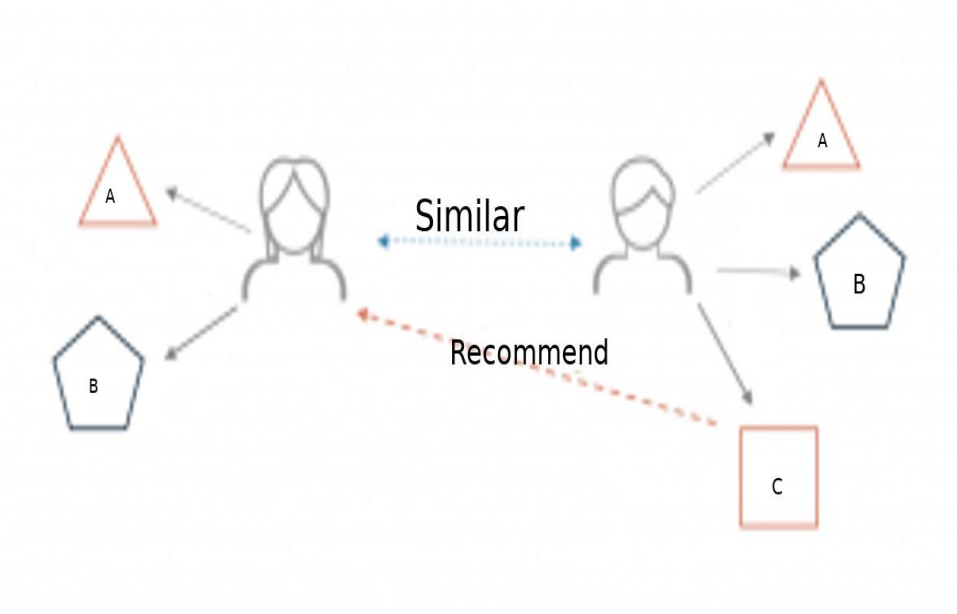


Figure 1.4. Working of Recommender system

1.1.2.1 Traditional Method Vs Recommender System

One can find the significance of these recommender system by comparing it with the traditional method of machine learning based recommendation systems. The old form of recommendation based system is related to the process of collaborative filtering. So, this method concerns taking note of the earlier recommendations of the users about a few things. Therefore, the previous ratings of the users about various systems is being considered in traditional systems. However, in the latest form of recommender system, one may find certain architectural structures which are operated on the basis of different neural networks. In order to understand the various artificial neural networks, one need to know that there is a recommendation system based on the system of deep learning.

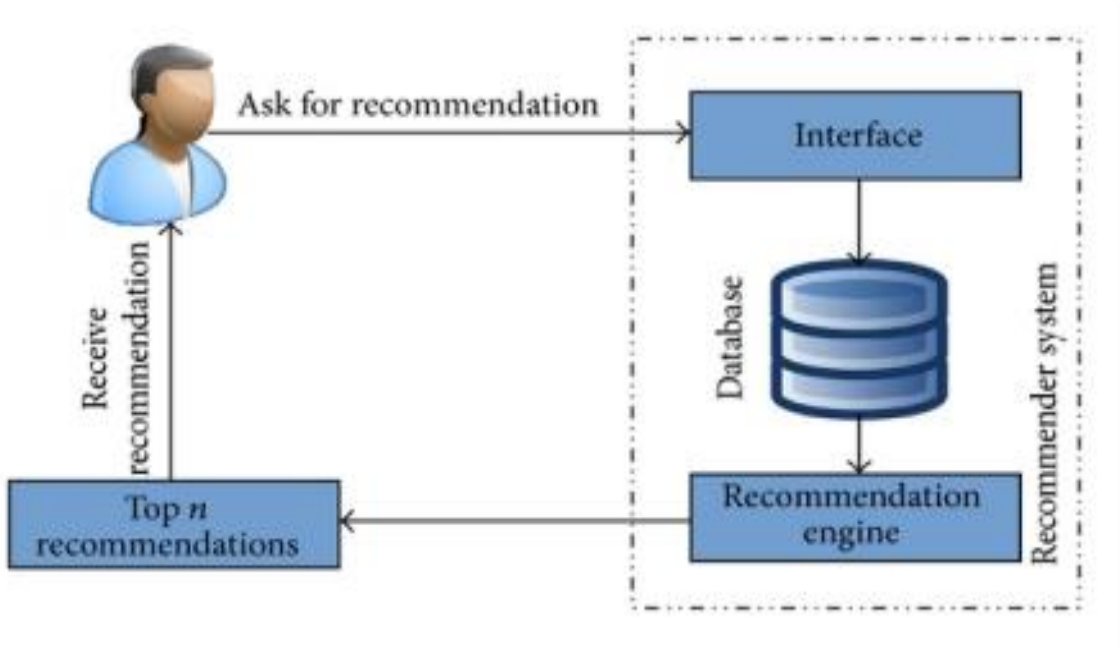


Figure 1.5. Traditional Recommendation System

In order to form an intelligent based system of recommendation, one need to find certain systems which can produce accurate results. There is an intelligent based system in the case of recommendation systems which can be used in this modern world. So, like a salesperson, this system works in order to offer various options to different users who can access these systems to fulfil various needs. In order to properly use these types of systems, it is important for us to understand that these systems are programmed to cater to user's preferences. So, all these recommendations are installed in a system which helps the computers to identify various forms of preferences which are appreciated by these users.

1.1.3 Types

There are basically four types of Recommender systems. They are content based, collaborative, knowledge based and hybrid recommender system.

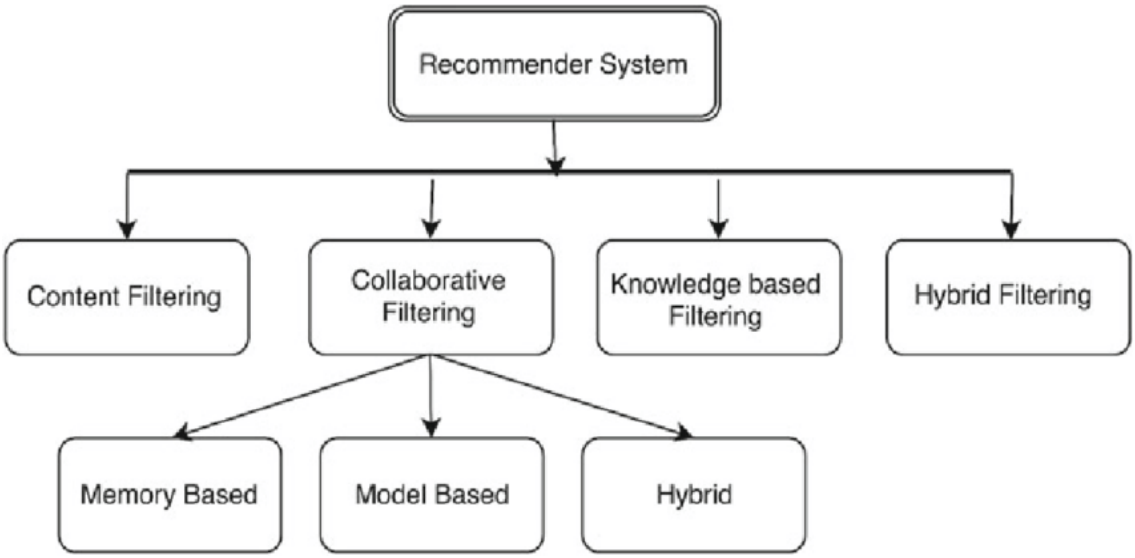


Figure 1.6. Types of Recommender systems

Collaborative or social-based were the first and are likely the most widely used type of recommendation system. Systems of this type seek to leverage ratings data from large numbers of users to find items of interest to recommend. The first methods utilized the k-nearest neighbor algorithm. These methods sought to find users and/or items with greatest similarity and use that additional information to make its predictions or recommendations.

This method requires the use of certain training models which makes this task easier. In order to find out a single parameter for implementing training model, one can take a look at the number of features in the dataset. The next step is to take into consideration the item's category where one does not need to label it, but to be aware of the item number.

Collaborative Filtering algorithms separated into two wide classes: memory based and model based algorithms.

Model Based Collaborative Filtering is based on offline mode. This technique compresses the complicated data set into lower dimensional set. It uses the Bayesian Network to capture relationship between user and item relation. Matrix factorization is another important technique that is first to point the QOS (Quality-of service) prediction problem.

Memory Based Collaborative Filtering Memory based collaborative filtering depends on an item to item or customer to customer similarity to mark estimation for customer on the items which customer has not used yet. If we put additional dataset into memory and added new data into existing set it reduces the performance of big dataset. This difficulty can be overcome by Recalculating correlations like Pearson correlation and vector correlation. Memory based collaborating filtering method are separated into two classes one is customer/user based collaborative filtering and another is item/product based Collaborative filtering.

Figure 1.7. Collaborative-based filtering

The content-based systems use similar methods as the collaborative filtering systems, but utilize a different set of data. The data used by content-based systems relies on features inherent to the item and not on ratings given to items by the users. Pandora serves as an excellent example of a content-based recommender. The method of content-based filtering is based on the presumption that there are a few keywords associated with the certain level of items. It can be observed from the system's working that these keywords tend to detect and identify the content which is related to such items. One can find many sort of recommendations in this form of algorithms which attempts to encourage various forms of recommendations. Use of a content-based has several advantages and disadvantages. It allows for significantly greater accuracy over a social-based model during a "cold start," the initial period of a recommender system that starts with no ratings. It requires developing a more compact representation of the content in addition to choosing features from analysis of the content to provide reduced computation expenses. Some common features include:

director, actors, genre (for movies), length, topic, author, word choice (for text), musician, lyrics, genre, key, and bpm (for music).

Figure 1.8. Content-based filtering

Knowledge-based systems attempt to learn rules and then use logic to make their recommendations. These systems work best in situations where ratings are sparse, due to the low frequency of their occurrence like house or car purchases, or where requirements need to be more precisely specified. There are two basic types of knowledge-based systems: constraint-based, that work by satisfying rules, and case-based, similarity metrics, system. The first might apply to home purchases. A prospective buyer specifies a price range and the systems works to provide them with available houses within that range. This type has a greater similarity to query-type systems than any other type of recommender systems. The second could be used in a local food finder that attempts to find nearby restaurants with food similar to other restaurants that the user has rated highly.

Hybrid recommender systems combine two or more recommendation strategies in different ways to benefit from their complementary advantages. These hybrid models could be as taking the weighted average of several models, or as complex as large monolithic systems that blur lines between models. This is a especially rich and useful topic to explore, as many of the best recommender systems get their edge from how they use models collectively. Learn about the various types of hybrid recommender systems here. It has been found through research that the combination of content-based filtering as well as a collaborative filtering results in the creation of a separate system. Such

system is known to be hybrid recommendation system which has both the features of content-based filtering and collaborative filtering system. Such sort of systems takes into consideration the likes of the users so that it can adjust the system to the level of preferences held by users.

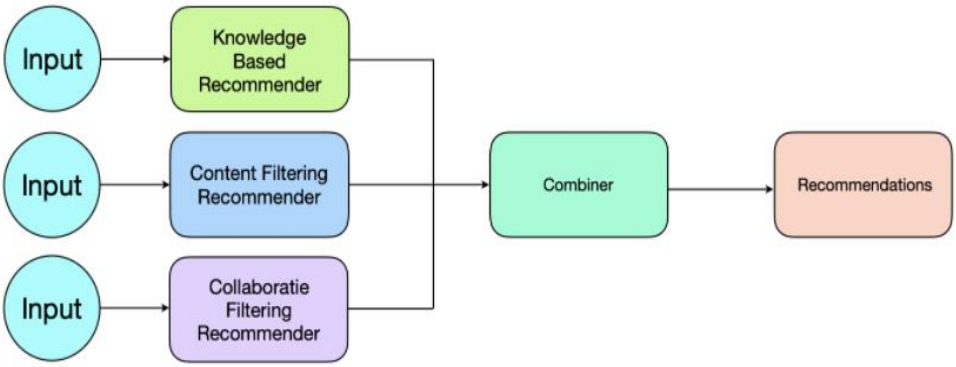


Figure 1.9. Hybrid recommender system

1.1.4 Uses

The most visible use of recommendation systems today comes in the form of commercial utilization. Amazon, Netflix, and Pandora are just a few of the companies that use recommendation systems in an effort to help customers/users find the items most relevant to them. Netflix uses your past ratings on movies to predict how much you will enjoy other movies. Pandora uses their thumb up/down ratings and content-based system to adjust the selection of songs played on each radio station. Amazon recommends related items based on past purchases, item views, and suggests related items that other customers purchased when viewing an item. These companies hope that by reducing the barrier to finding what their customers want they can increase sales, satisfaction, and stability in the consumer base. Another benefit of using such form of systems consist of achieving a level of customer satisfaction. So, this is probably one of the best form of uses one can get after taking care of these systems. This system has the ability of personalizing things so that the people who are accessing this form of data can recommend different things to these users. Therefore, such systems know the data about yourself and like your family and friends help in recommending things to you.

1.2 Problem Definition

The world of retail is changing rapidly. Many brick and mortar locations are closing and being replaced by online stores. A lot of eCommerce platforms fail to sell through a high percentage of their merchandise. This is often due to poor user browsing experience. Shoppers need to be provided suggestions based on their likes and needs in order to create a better shopping environment that boosts sales and increases the time spent on a website.

Recommender engine is mostly used to aid in consumer decision making. Obviously many e-retailers like Amazon have already been using recommender algorithms, but many smaller or newer sites are still in need.

1.3 Objective of Project

The main Objective of this project is to recommend the related products for the particular product based on product-id in product-based filtering and in user based filtering the similar users are identified and the products are recommended appropriately.

1.4 Organization of Documentation

The Organization of documentation as follows:

Chapter 1: Describes the Introduction of Project, Objective.

Chapter 2: Describes the Literature Survey.

Chapter 3: Describes the Analysis.

Chapter 4: Describes the Design.

Chapter 5: Describes the Methods.

Chapter 6: Describes the Implementation & Results.

Chapter 7: Describes the Testing & Validation.

CHAPTER-2

LITERATURE SURVEY

2.1 Introduction

Currently, we are facing an ever growing deluge of information, music, videos, books, and other commercial merchandise. While this provides new opportunities to find goods tailored to our needs or preferences, the overall abundance serves as an increasing barrier to find items of interest or relevance. Additionally, high visibility is given to only the small fraction of items that are popular across diverse groups. This small fraction of items account for a large percentage of sales and has the effect of creating a long tail in the distribution, lots of items that individually have small sales volume, but together still constitute a significant percentage of total sales. The long tail distribution in both sales and visibility provides a hindrance to the discovery of items most relevant to a consumer's needs. Recommendation system is meant for an approach to get help in decision making, specifically for those users so-far experiencing the complex type of information handling environment.

Recommendation systems have become an increasingly prevalent presence in our lives as our choices in music, videos, books, and even household goods expand. Companies like Netflix, Pandora, and Amazon look to harness the power of recommendation systems to increase their bottom line. However, they also represent an opportunity for users/customers to find relevant items without wasting precious time wading through the wide array of choices available to them.

2.2 Existing System

Collaborative filtering is again classified into two types, model based and memory based. In memory based there are product based and user based collaborative filtering. Based on the type of application the appropriate collaborative filtering is used. Product based filtering is mostly used for e-commerce and user based filtering is used for movie recommendations.

- Recommender systems take advantage of several sources of information to predict the preferences of users for items of interest [1].
- Collaborative filtering is one of the most widely used algorithm for product recommendation, and it is considered effective[2].
- Memory-based CF is an early generation CF that uses heuristic algorithms to calculate similarity values between users or items, and can therefore be subdivided into two types: user-based CF and item-based CF [3].
- Item based collaborative filtering finds similarity patterns between items and recommends them to users based on the computed information, whilst user based finds similar users and gives them recommendations based on what other people with similar consumption patterns appreciated[4].
- The CF approaches use statistical techniques to analyze the similarity between users and to form a set of users called neighbors. A set of similarity measures is a metric of relevance between two vectors [5]
- The earliest approaches would now be classified as collaborative filtering, they sought to find similar users to the questor and utilize the data available to build personal filters [6]
- The CF approaches use statistical techniques to analyze the similarity between users and to form a set of users called neighbors. A set of similarity measures is a metric of relevance between two vectors [7]
- Since the similarity measure plays a significant role in improving accuracy in prediction algorithms, it can be effectively used to balance the ratings significance [8]

2.3 Proposed System

Our approach is to develop a recommendation system by using both product based and user based collaborative filtering for the same dataset. For this we are using K-Nearest Neighbor (KNN) algorithm. The accuracy is compared between the product-based and user-based filtering.

CHAPTER-3

ANALYSIS

3.1 Introduction

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches .The most critical section of the project plan is a listing of high level product requirements, also referred to as goals .All of the software product requirements to be developed during the requirements definition stage flow from one or more of the these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

3.2 Software Requirements Specification

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system yow more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software is initiated by the client needs. The SRS is the means of translating the ideas of the minds of the clients (the input) into a formal document (the output of the requirement phase).

3.2.1 Hardware Requirements

- Any Contemporary PC

3.2.2 Software Requirements

Operating System	:	Any Windows OS
Tools used	:	Jupyter Notebook

Dataset	:	CSV file
Languages used	:	Python

3.2.3 Python Installation Process

Python is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.

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

The Python download requires about 25 Mb of disk space; keep it on our machine, in case we need to re-install Python. When installed, Python requires about an additional 90 Mb of disk space.

Downloading

1. First Go to the Python Downloads Site. The following page will appear in your browser.

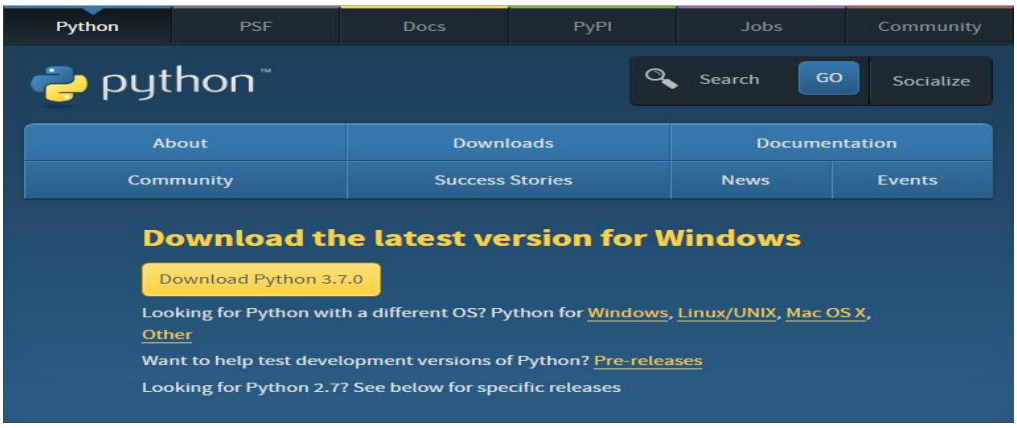
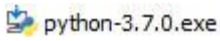


Figure 3.1. Python Downloads Site

2. Click the **Download Python 3.7.0** button.

The file named **python-3.7.0.exe** should start downloading into our standard download folder. This file is about 30 Mb so it might take a while to download fully if we are on a slow internet connection. The file should appear as



3. Move this file to a more permanent location, so that we can install Python (and reinstall it easily later, if necessary).
4. Feel free to explore this webpage further; if we want to just continue the installation, we can terminate the tab browsing this webpage.
5. Start the **Installing** instructions directly below.

Installing

1. Double-click the icon labeling the file **python-3.7.0.exe**.

   An **Open File - Security Warning** pop-up window will appear.

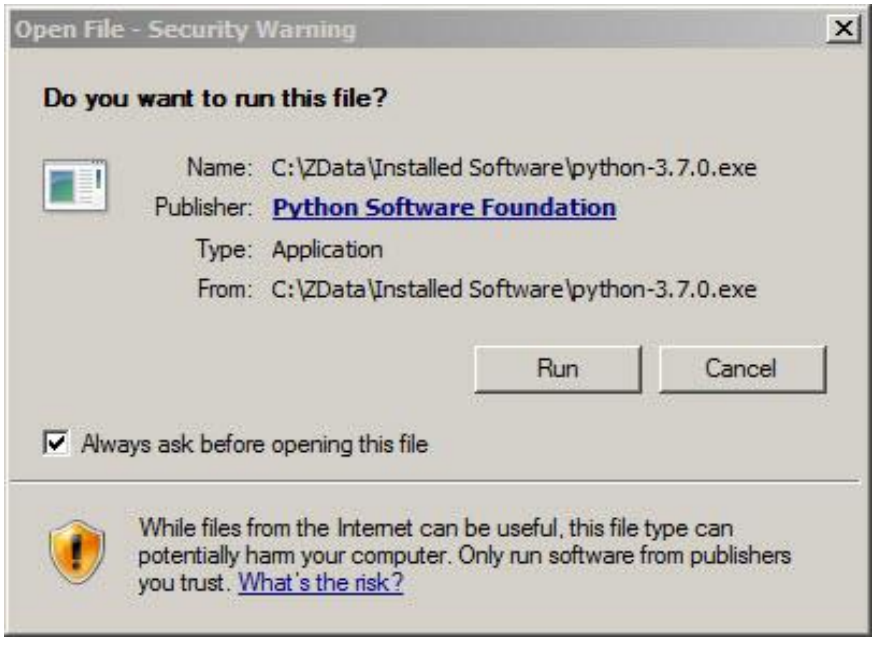


Figure 3.2. Open File Pop-up Window

2. Click **Run**. A **Python 3.7.0 (32-bit) Setup** pop-up window will appear.

Figure 3.3. Python Setup Window

Ensure that the **Install launcher for all users (recommended)** and the **Add Python 3.7 to PATH** checkboxes at the bottom are checked.

If the Python Installer finds an earlier version of Python installed on our computer, the **Install Now** message may instead appear as **Upgrade Now** (and the checkboxes will not appear).

3. Highlight the **Install Now** (or **Upgrade Now**) message, and then click it.

   A **User Account Control** pop-up window will appear, posing the question **Do you want the allow the following program to make changes to this computer?**

Figure 3.4. User Account Control

4. Click the **Yes** button.

A new **Python 3.7.0 (32-bit) Setup** pop-up window will appear with a **Setup Progress** message and a progress bar.

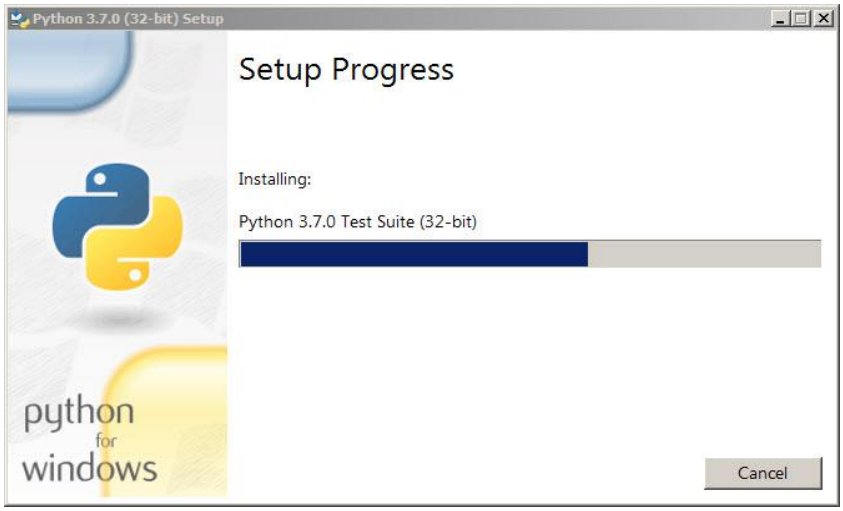


Figure 3.5. Installing Python

During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new **Python 3.7.0 (32-bit) Setup** pop-up window will appear with a **Setup was successful** message.

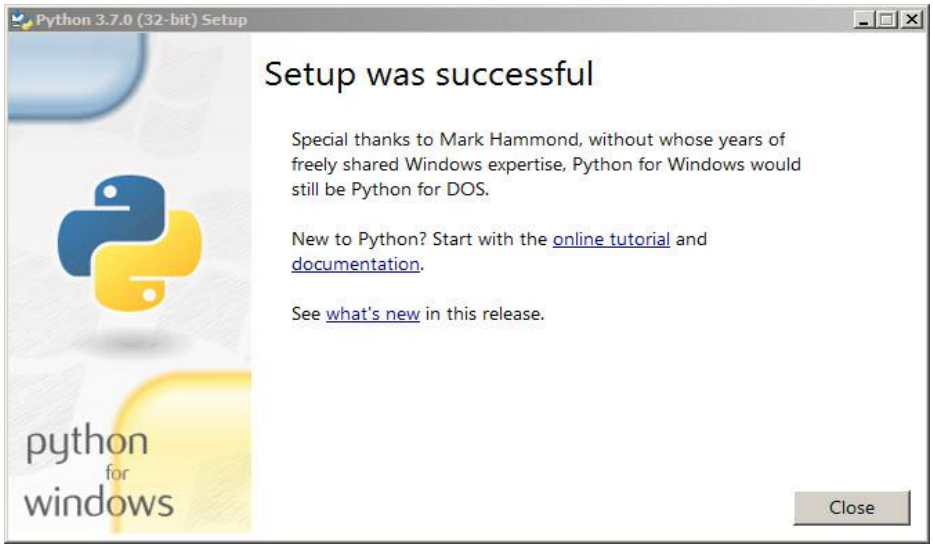


Figure 3.6. Python Setup Successful

5. Click the **Close** button.

Python should now be installed.

Verifying

To try to verify installation,

1. Navigate to the directory **C:\Users\Pattis\AppData\Local\Programs\Python\Python37-32** (or to whatever directory Python was installed: see the pop-up window for Installing step 3).

2. Double-click the icon/file **python.exe**.

   The following pop-up window will appear.

Figure 3.7. Verifying Installation

A pop-up window with the title **C:\Users\Pattis\AppData\Local\Programs\Python\Python37-32** appears, and inside the window; on the first line is the text **Python 3.7.0 ...** (notice that it should also say 32 bit). Inside the window, at the bottom left, is the prompt >>>: type **exit()** to this prompt and press **enter** to terminate Python.We should keep the file **python-3.7.0.exe** somewhere on our computer in case we need to reinstall Python (not likely necessary).

We may now follow the instructions to download and install Java (we should have already installed Java, but if we haven't, it is OK to do so now, so long as we install both Python and Java before we install Eclipse), and then follows the instruction to download and install the Eclipse IDE. We need to download/install Java even if we are using Eclipse only for Python.

Install Pip:

pip is a package-management system written in Python used to install and manage software packages. It connects to an online repository of public packages, called the Python Package Index.

Pip is a Package manager for python which we will use to load in modules/libraries into our environments. An example of one of these libraries is VirtualEnv which will help us keep our environments clean from other Libraries. To test that Pip is installed open a command prompt (win+r→'cmd'→Enter) and try 'pip help'.

3.2.4 Jupyter Notebook Installation

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text . Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter notebook remains one of the best editors. Jupyter is a user-friendly editor that is great for data analysis on account of its clear background and fast as well as impressive display.

Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name , Jupyter, comes from the core supported programming languages that it supports:Julia, Python and R. Jupyter ships with the IPython kernel, which allows you to write your programs in python, but there are currently over 100 other kernels that you can also use. IPython notebook was developed by Fernando Perez as a web based affront end to IPython kernel.

A notebook document consists of rich text elements with HTML, formatted text, figures, mathematical equations etc. The notebook is also an executable document consisting of code blocks in python or other supporting languages.

Jupyter notebook is a client-server application. The application starts the server on a local machine and opens the notebook interface in the web browser where it can be edited and run from. The notebook is saved as an ipynb file and can be exported as html, pdf and LaTex files.

Getting Up and Running With Jupyter Notebook

The Jupyter Notebook is not included with Python, so if you want to try it out, you will need to install Jupyter.

There are many distributions of the Python language. This article will focus on just two of them for the purposes of installing Jupyter Notebook. The most popular is CPython, which is the reference version of Python that you can get from their website. It is also assumed that you are using Python 3.

Installation

To install Jupyter Notebook ensure that you tick “Add Python to path” when installing Python.

Then go to computer’s Command Prompt. To find Command Prompt, type “cmd” in the “Type here to search” taskbar at the bottom left of Windows computer. Open the Command Prompt, type the command below, then press enter:

python -m pip install -upgrade pip

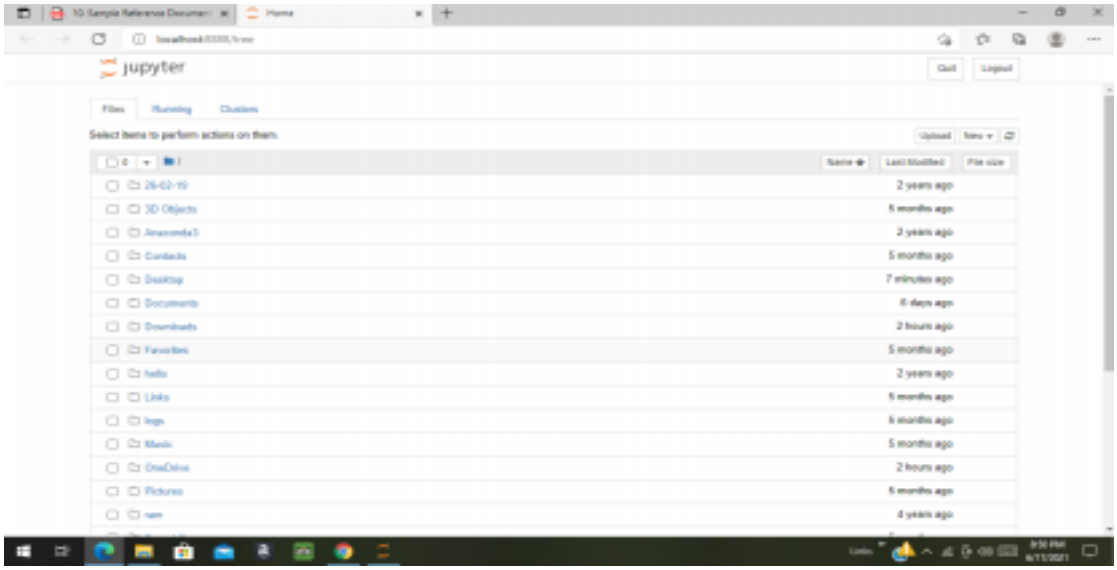
The command above will upgrade pip if not upgraded already. After upgrading pip, write the command below inside the Command Prompt and press enter:

python -m pip install jupyter

Wait for Jupyter Notebook to install. Once the installation is completed the computer will display a “successfully installed” message. To open your Jupyter notebook, write the command below inside the Command Prompt and press enter:

jupyter notebook

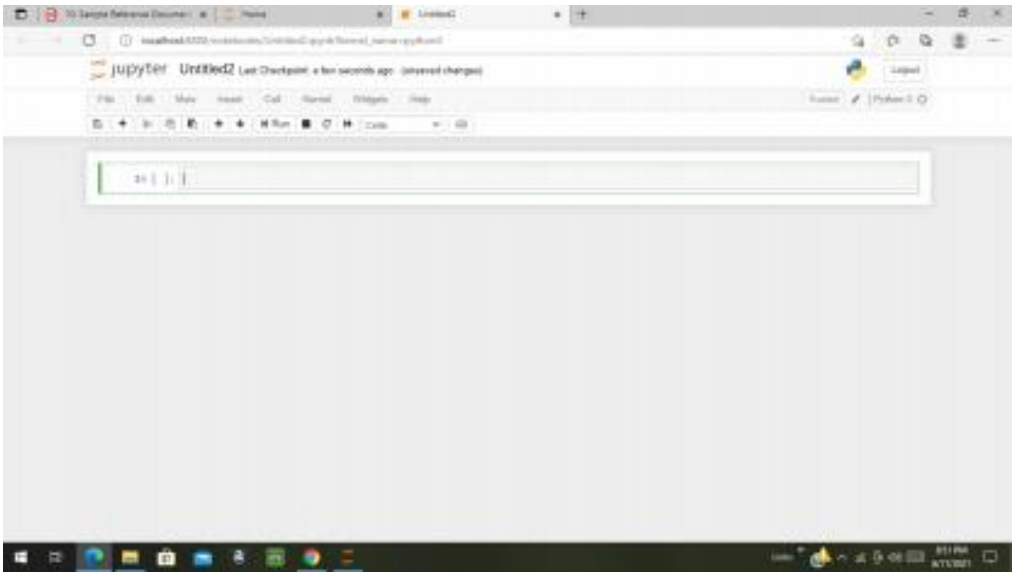
Jupyter Notebook is automatically opened in one of the browsers.



Screen 3.1 Jupyter notebook

Creating a NoteBook

Now that you know how to start a Notebook server, you should probably learn how to create an actual Notebook document. All you need to do is click on the New button (upper right), and it will open up a list of choices.



Screen 3.2 Notebook of Jupyter Notebook

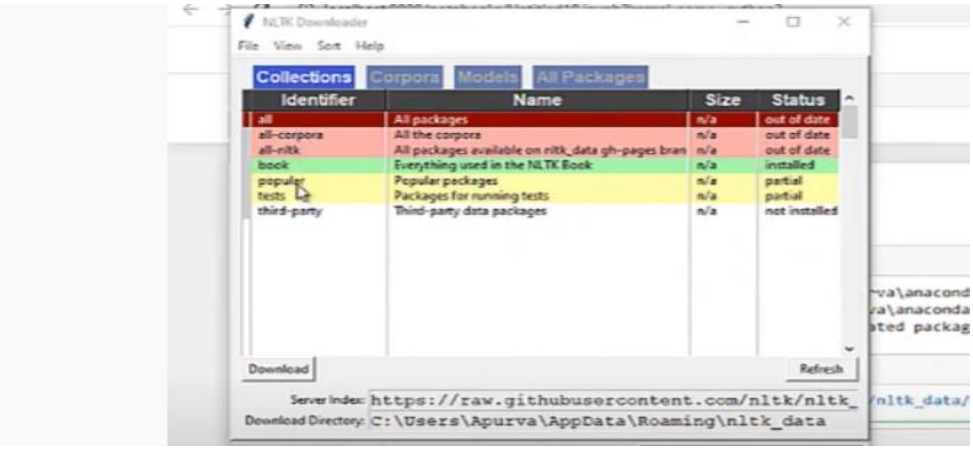
3.2.5 Installation of NLTK

The Natural Language Toolkit (**NLTK**) is a platform used for building **Python** programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. It also includes graphical demonstrations and sample data sets as well as accompanied by a cook book and a book which explains the principles behind the underlying language processing tasks that NLTK supports.

NLTK has been called "a wonderful tool for teaching, and working in, computational linguistics using Python," and "an amazing library to play with natural language."

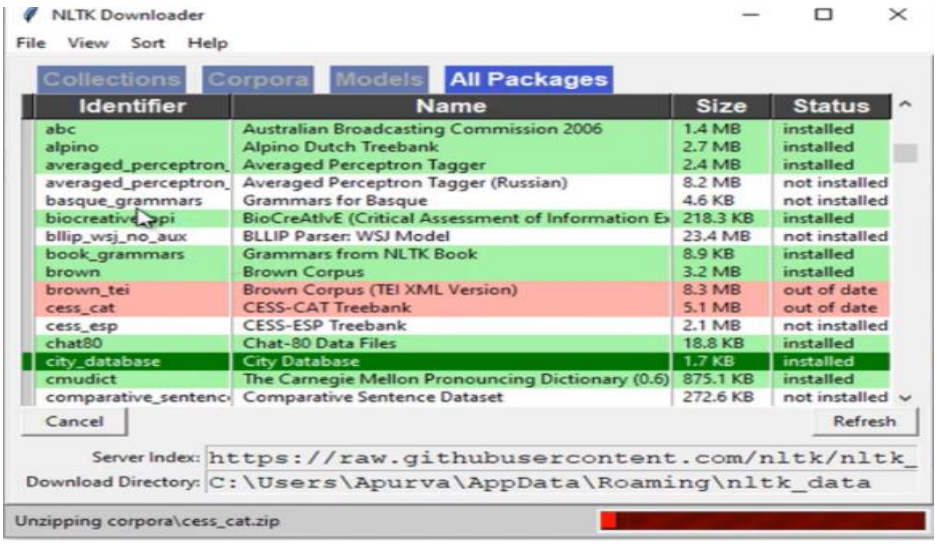
Installation

The installation of NLTK through jupyter Notebook is simple. To install NLTK through Jupyter Notebook , open the notebook and type the command **pip install nltk** Then click Enter. Now type the command **nltk.download()** and click Enter. Now a window will be popped up where we can download different collections, corpra, models and Allpakages that are required for the project.



Screen 3.3 NLTK Downloader

Click on download button in the pop up screen.



Screen 3.4 Downloading NLTK

Once the installation is completed nltk packages can be imported.

CHAPTER-4

DESIGN

4.1 Introduction

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business processes diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The output of the design stage are the design document, an updated RTM, and an updated project plan.

4.2 UML Diagrams

4.2.1 Unified Modeling Language (UML)

The Unified Modeling Language is a standard language for specifying visualizing, constructing and documenting the software system and its components. It is a graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain and control information about the systems.

Visualizing: Through UML we view an existing system and ultimately we visualize how the system going to be after implementation unless we think we cannot implement.

UML helps to visualize, how the components of the system communicate and interact with each other.

Specifying:

Specifying means building models that are precise, unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

Documenting:

The Deliverables of a project apart from coding are some Artifacts, which are artificial in controlling, measuring and communicating about a system during its development viz, requirements, architecture, desire, source code, project plans, tests, prototypes, releases etc.

4.2.2 Use case Diagrams

These diagrams show a set of use cases and actors and their relationships. These diagrams illustrate the static use case view of a system and are important in organizing and modeling the behaviors of a system. The Use case diagram is used to identify the primary elements and processes that form the system. The primary elements are termed as “actors” and the processes are called “use cases”. The Use case diagram shows which actors interact with each use case.

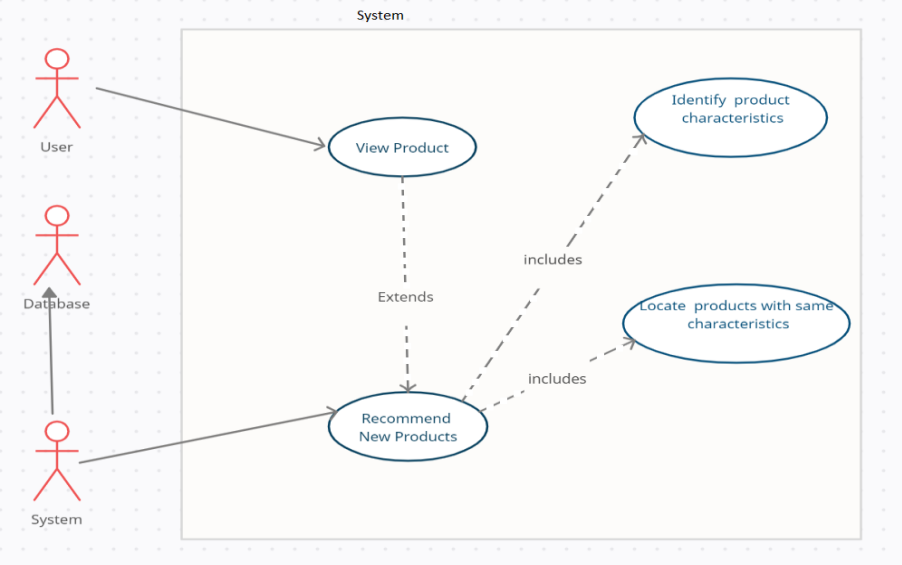


Figure 4.1. Use case Diagram for product-based filtering

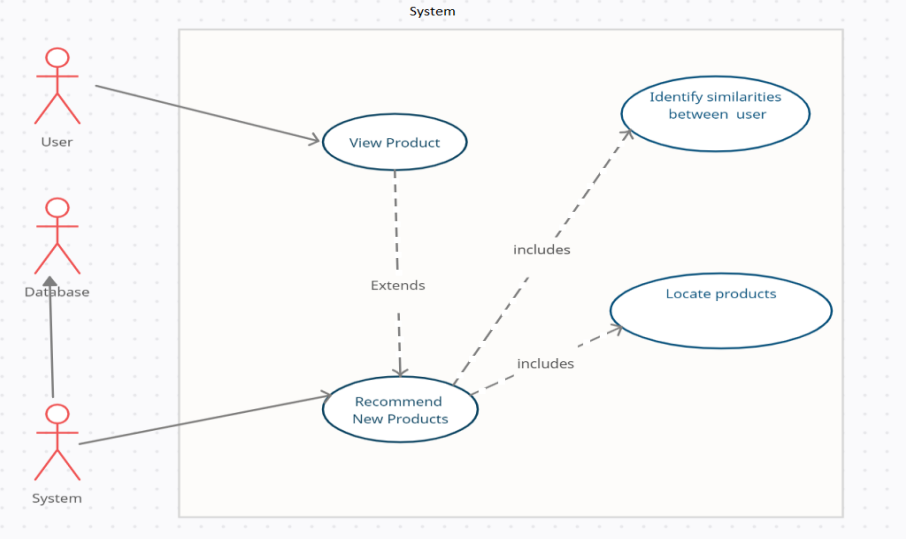


Figure 4.2. Use case Diagram for user-based filtering

4.3 Data Preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

4.4 Processes of Recommender Systems

Recommender systems are essential for web-based companies that offer a large selection of products. Those recommender systems provide value to customers by understanding an individual user's behaviour and then recommending to them items they might find useful. There are a series of steps which are followed in order to undertake these processes about recommender systems. These steps are about collection, storing, analysis and filtering of data which can be done step wise. These steps are mentioned below. All of these processes are used to ensure that the people are accessing the right kind of data which help them gain recommendation through such

systems. It makes them discover and explore various forms of data which can be used to boost sales of their businesses also.

4.4.1 Collecting the data

In this step, the machine learning system uses data which is available and present in different forms, whether explicit or implicit in nature. The data which is termed explicit can be collected by looking at the reviews and opinion sharing of the users about various products. However, the data of implicit nature is related to the search log and history of the data which is accessed in a different form. This data can be accessed easily by keeping a look at the search log or the history of users which notes down or collect the record about user's preferences.

4.4.2 Storing of data

This steps involves storing the data which has been collected earlier or it involves saving the data in a system which keeps giving the recommendations later. Therefore, the storage of the given data helps in bringing out the recommendations which can be made about the system and can encourage facilitation of recommendations about users.

4.4.3 Analysing the data

After going through the above given steps, one may find the data moving toward the analysis stage in which data is analysed thoroughly. The systems used for the data analysis include real-time systems, batch analysis and near-real-time analysis form of data systems.

4.4.4 Filtering of data

This step involves giving recommendations to the users after carefully analysing the data which has to be filtered later on.

CHAPTER-5

METHODS

5.1 Collaborative Filtering

Collaborative filtering is one of the well known and most extensive techniques in recommendation system its basic idea is to predict which items a user would be interested in based on their preferences. Recommendation systems using collaborative filtering are able to provide an accurate prediction when enough data is provided, because this technique is based on the user's preference.

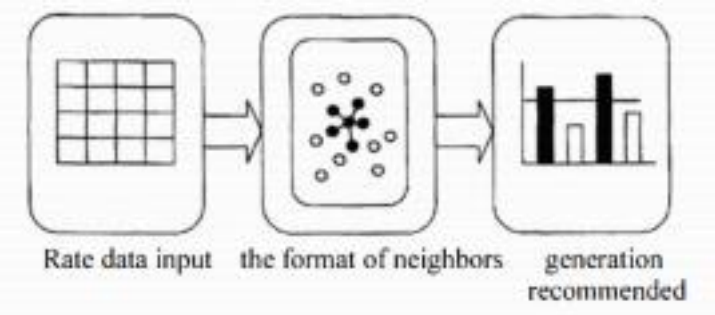


Figure-5.1. Collaborative filtering

Collaborative filtering is an idea that is related to crowd sourcing. The basic idea is the use of large numbers of users and ratings to find similar items and users that can assist in the creation of predictions. Similarity measures are functions for determining how much one items is like another given a vector of features that describes them. Common similarity measures include cosine similarity and distance functions (Manhattan, Euclidean, Minkowski). For increased performance, item-item similarity is often used in conjunction with caching due to the lower volatility of their similarity measures. Requires a fairly significant number of ratings before any level of accuracy is guaranteed, however the accuracy of the systems will increase over time as more ratings, users, and items enter the system. New users and items need a certain number of ratings (items more so than users) before accurate predictions can be made even if the rest of the systems has achieved a higher level of accuracy.

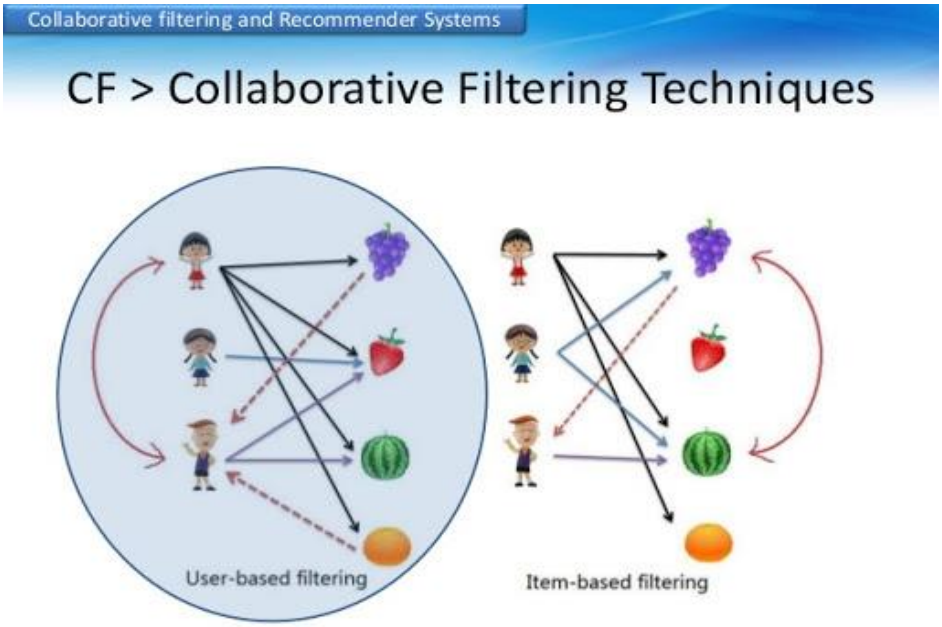


Figure 5.2. User-based and Item-based filtering

Now that most of the bias has been eliminated from the ratings data, another approach to predicting ratings must be found if greater improvement is sought. One approach is to find similar users that have rated the item and use those ratings in our prediction. However, that raises the question of determining similarity between users. In collaborative filtering we ignore the User and Item attributes. We focus on User-Item interactions to recommend the products.

5.2 Similarity Measures

Similarity in a recommender system is about finding items (or users, or user and item) that are similar. Recommendation system is one of the most valuable approaches to provide personalized services for users. It helps in finding the relevant information as per user's interest from enormous amount of data. To achieve this, a similarity measure is used that computes the similarity between two users or items. There are multifarious methods to compute the similarity between users/items, but each method has some limitations.

Different similarity measure algorithms used are Pearson correlation, Euclidean distance, Manhattan distance, Minkowski distance, Cosine similarity and Jaccard coefficient. The algorithms used in this paper behave differently

in different context. Majority of the algorithms showed the same result in finding the similarity between the users.

For the project Euclidean distance is used. The Euclidean distance between two points is the length of the line segments connecting them. Our Euclidean space in this particular case is the positive portion of the plane where the axes are the ranked items and the points represent the scores that a particular person gives to both items.

Euclidean distance method is based on the distance between items. It forms coordinates to put preference values between items and measures Euclidean distance between each point. When distance value between two points is large, it means the two points are not similar. When distance value between two points is small, it means two points are similar.

5.3 K-Nearest Neighbor

K-Nearest Neighbour is one of the simplest Machine Learning algorithms. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. KNN finds the clusters of similar users/products based on common ratings, and make predictions using the average rating of top-k nearest neighbors.

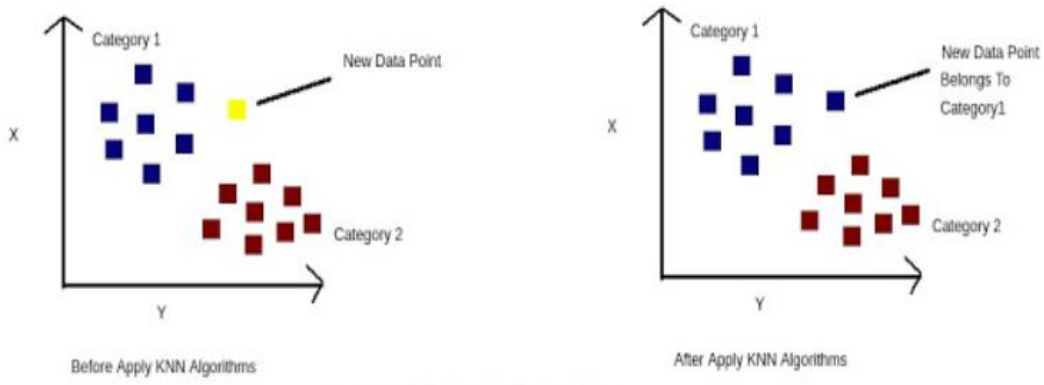


Figure 5.3. Working of KNN Algorithm

K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time

of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Algorithm:

- Step-1: Select the number K of the neighbors
- Step-2: Calculate the Euclidean distance of K number of neighbors
- Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step-4: Among these k neighbors, count the number of the data points in each category.
- Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
- Step-6: Our model is ready.

5.3.1 Product based filtering

Item-item collaborative filtering, or item-based, or item-to-item, is a form of collaborative filtering for recommender systems based on the similarity between items calculated using people's ratings of those items. Item-item collaborative filtering was invented and used by Amazon.com in 1998. Item-item models resolve the problems in systems that have more users than items. Item-item models use rating distributions per item, not per user. Item-based collaborative filtering approach is to predict items by inquiring into similarities between the items and other items that are already associated with the user.

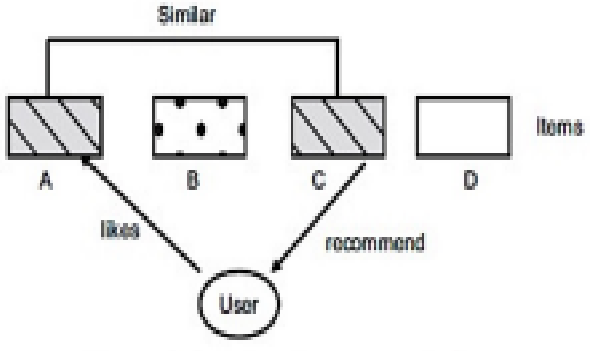


Figure 5.4. Product based filtering

For example, as shown in above figure 5.3, let's say Item A and Item C are very similar. If a User likes Item A, IBCF can recommend Item C to the User. IBCF needs a set of items that the target user has already rated to calculate similarities between items and a target item. And then, it generates prediction in terms of the target item by combining the target user's previous preferences based on these item similarities .

In IBCF, users' preference data can be collected in two ways. One is that user explicitly gives rating score to item within a certain numerical scale. The other is that it implicitly analyzes user's purchase records or click-through rate.

With more users than items, each item tends to have more ratings than each user, so an item's average rating usually doesn't change quickly. This leads to more stable rating distributions in the model, so the model doesn't have to be rebuilt as often. When users consume and then rate an item, that item's similar items are picked from the existing system model and added to the user's recommendations.

In this method ,first, the system executes a model-building stage by finding the similarity between all pairs of items. This similarity function can take many forms, such as correlation between ratings or cosine of those rating vectors. As in user-user systems, similarity functions can use normalized ratings.

Second, the system executes a recommendation stage. It uses the most similar items to a user's already-rated items to generate a list of recommendations. Usually this calculation is a weighted sum or linear regression.

Contrarily to user-based methods, item similarity matrices tend to be smaller, which will reduce the cost of finding neighbours in our similarity matrix. Also, since a single item is enough to recommend other similar items, this method will not suffer from the cold-start problem. A drawback of item-based methods is that they there tends to be a lower diversity in the recommendations as opposed to user-based CF.

5.3.2 User based filtering

User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user. The main idea behind UBCF is that people

with similar characteristics share similar taste. User-based collaborative filtering approach is to predict items to the target user that are already items of interest for other users who are similar to the target user.

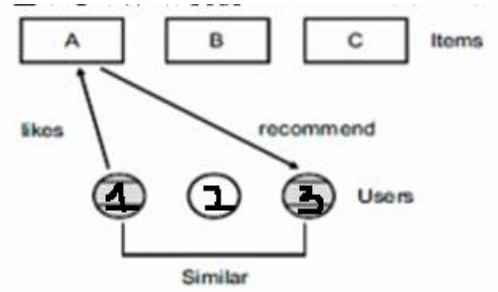


Figure 5.5. User-based filtering

For example, as shown in above figure 5.4, let User 1 and User 3 have very similar preference behaviour. If User 1 likes Item A, UBCF can recommend Item A to User 3. UBCF needs the explicit rating scores of items rated by users to calculate similarities between users and exploits k-nearest neighbor algorithms to find the nearest neighbors based on user similarities. And then, it generates prediction in terms of items by combining the neighbor user's rating scores based on similarity weighted averaging.

The algorithm we use for user based collaborative filtering begins with the set of training data after filtering for users that have a rating for the item we are attempting to predict. We then compute similarity, between the remaining users and the user for whom we are attempting to predict.

A drawback is that there tends to be many more users than items, which leads to much bigger user similarity matrices (this might be clear in the following section) leading to performance and memory issues on larger datasets, which forces to rely on parallelisation techniques or other approaches altogether.

CHAPTER-6

IMPLEMENTATION & RESULTS

6.1 Libraries used

Python is increasingly being used as a scientific language. Matrix and vector manipulation are extremely important for scientific computations. Both NumPy and Pandas have emerged to be essential libraries for any scientific computation, including machine learning, in python due to their intuitive syntax and high-performance matrix computation capabilities.

NumPy:

NumPy stands for 'Numerical Python' or 'Numeric Python'. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem. NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation. NumPy can be imported into the notebook using: **import numpy as np.**

Pandas:

Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Pandas provides an in-memory 2d table object called Data frame. It is like a spreadsheet with column names and row labels. Hence, with 2d tables, pandas are capable of providing many additional functionalities like creating pivot tables, computing columns based on other columns and plotting graphs. Pandas can be imported into Python using:

import pandas as pd.

Regular Expression(re):

A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing). Regular Expressions can be imported using: **import re**

NLTK:

The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities. Natural Language Toolkit can be imported into python using: **import nltk.**

Sklearn:

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. In our project we have used different features of sklearn library like:

from sklearn.neighbors import NearestNeighbors

NearestNeighbors implements unsupervised nearest neighbors learning. It acts as a uniform interface to three different nearest neighbors algorithms: BallTree, KDTree, and a brute-force algorithm based on routines in sklearn.metrics.pairwise. The choice of neighbors search algorithm is controlled through the keyword ‘algorithm’, which must be one of ['auto’,’ball_tree’,’kd_tree’,’brute’]. When the default value ‘auto’ is

passed, the algorithm attempts to determine the best approach from the training data. In this project the BallTree algorithm is used.

from sklearn.metrics import classification_report

A Classification report is used to measure the quality of predictions from a classification algorithm. The report shows the main classification metrics precision, recall and f1-score on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives. The reported averages include macro average (averaging the unweighted mean per label), weighted average (averaging the support-weighted mean per label), and sample average (only for multilabel classification). Micro average (averaging the total true positives, false negatives and false positives) is only shown for multi-label or multi-class with a subset of classes, because it corresponds to accuracy otherwise and would be the same for all metrics.

from sklearn.metrics import accuracy_score

In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must exactly match the corresponding set of labels in y_true.

from sklearn.feature_selection import SelectKBest

The classes in the **sklearn.feature_selection** module can be used for feature selection/dimensionality reduction on sample sets, either to improve estimators' accuracy scores or to boost their performance on very high-dimensional datasets. The SelectKBest method selects the features according to the k highest score. By changing the 'score_func' parameter we can apply the method for both classification and regression data. Selecting best features is important process when we prepare a large dataset for training.

from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer

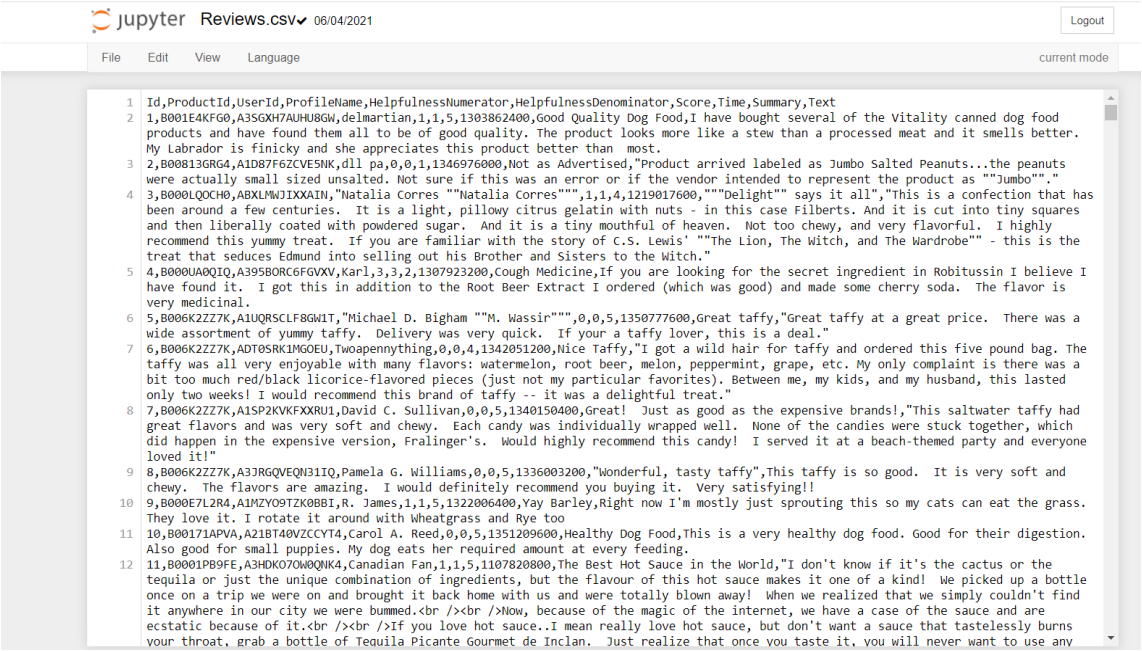
The sklearn.feature_extraction module can be used to extract features in a format supported by machine learning algorithms from datasets consisting of formats such as text and image. **CountVectorizer** is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the

frequency (count) of each word that occurs in the entire text. **TF-IDF** is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction.

CSV file:

The dataset used in this project is a .CSV file.

In computing, a comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. CSV is a simple file format used to store tabular data, such as a spreadsheet or database. Files in the CSV format can be imported to and exported from programs that store data in tables, such as Microsoft Excel or OpenOffice Calc. Its data fields are most often separated, or delimited, by a comma.



Screen 6.1 Dataset with 10 attributes

A CSV is a comma-separated values file,which allows data to be saved in a tabular format. CSVs look like a garden-variety spreadsheet but with a .csv extension. CSV files can be used with most any spreadsheet program, such as Microsoft Excel or Google Spreadsheets.

The difference between CSV and XLS file formats is that CSV format is a plain text format in which values are separated by commas (Comma Separated Values), while XLS file format is an Excel Sheets binary file format which holds information about all the worksheets in a file, including both content and formatting.

6.2 Implementation

All the required libraries like Sklearn and Modules like Numpy , Pandas , Re , nltk are imported into the Jupyter notebook initially into the file created in the notebook. After importing all the modules and libraries into the notebook , A csv file has to be loaded using Pandas into the notebook.The implementation of these will be as follows:

```
import numpy as np
import pandas as pd
from pandas import DataFrame
import nltk
from sklearn.neighbors import NearestNeighbors
from sklearn import neighbors
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.feature_selection import SelectKBest
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
import re
```

```
df = pd.read_csv("Reviews.csv")
```

Screen 6.2 Importing modules and libraries

DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. It is similar to spreadsheet or SQL table, or a dict of Series objects. It is generally the most commonly used pandas object. The Dataframes are used inorder to form a 2D array with the appropriate attributes that are required.

Regular expression module is used to remove the spaces and special symbols i.e., to perform clean up operation.

6.2.1 Product-based filtering

In product-based filtering the similarity between the products is calculated and the three most similar products are recommended. The products are uniquely identified by the product-id. The products in the dataset are grouped by using product-id and the count is calculated. The products with the count greater than or equal to 100 are formed into a dataframe and the mean value of score is calculated.

```
count = df.groupby("ProductId", as_index=False).count()
df1 = pd.merge(df, count, how='right', on=['ProductId'])
df1 = df1[['ProductId','Summary','Score',"Count"]]
df1 = df1.sort_values(['Count'], ascending=False)
df2 = df1[df1.Count >= 100]
df4 = df.groupby("ProductId", as_index=False).mean()
```

Screen 6.3 Grouping by product-id and calculating mean

The summaries of each product is combined with the product-id's and the clean up process is performed on the summary.

```
combine_summary = df2.groupby("ProductId")["Summary"].apply(list)
combine_summary = pd.DataFrame(combine_summary)
combine_summary.to_csv("combine_summary.csv")
df3 = pd.read_csv("combine_summary.csv")
df3 = pd.merge(df3, df4, on="ProductId", how='inner')
df3 = df3[['ProductId','Summary','Score']]
cleanup_re = re.compile('[^a-z]+')
def cleanup(sentence):
    sentence = sentence.lower()
    sentence = cleanup_re.sub(' ', sentence).strip()
    sentence = " ".join(nltk.word_tokenize(sentence))
    return sentence
df3["Summary_Clean"] = df3["Summary"].apply(cleanup)
```

Screen 6.4. Combining and clean-up of summary for PBCF

Now the duplicates from the dataframe are removed on the basis of score except the last tuple and then reset the index.

```
df3 = df3.drop_duplicates(['Score'], keep='last')
df3 = df3.reset_index()
```

Screen 6.5. Removing duplicates and reset the index for PBCF

From the summary obtained by the above step 100 features of the were extracted are transformed into vectors and a dataframe is created with the indices similar to summary_clean.

```
docs = df3["Summary_Clean"]
vect = CountVectorizer(max_features = 100, stop_words='english')
X = vect.fit_transform(docs)
df5 = DataFrame(X.A, columns=vect.get_feature_names())
```

Screen 6.6. Extracting features and vector transformation

To create training and test data the dataset obtained by the above process is divided accordingly.

```
X = np.array(df5)
tpercent = 0.9
tsize = int(np.floor(tpercent * len(df5)))
df5_train = X[:tsize]
df5_test = X[tsize:]
lentrain = len(df5_train)
lentest = len(df5_test)
```

Screen 6.7. Division of training and test data for PBCF

Next a nearest neighbor object has to be instantiated, and call it nbrs Then fit it to dataset X. To find the k-neighbors of each point in object X, call the kneighbors() function on object X and “Ball_Tree” algorithm is used to find three nearest neighbors.

```
nbrs = NearestNeighbors(n_neighbors=3, algorithm='ball_tree').fit(df5_train)
distances, indices = nbrs.kneighbors(df5_train)
```

Screen 6.8. Instantiating object for PBCF

Then the similar products for the test dataset is found and then the accuracy, precision, recall, F1 score, support are calculated.

6.2.2 User-based filtering

In user-based filtering the similarity between the users are calculated and the products purchased by similar user are recommended. The users are uniquely identified by the user-id. The users in the dataset are grouped by using user-id and the count is calculated. The users with the count greater than or equal to 100 are formed into a dataframe and the mean value of score is calculated.

```
count = df.groupby("UserId", as_index=False).count()
df1 = pd.merge(df, count, how='right', on=["UserId"])
df1 = df1.sort_values(['Count'], ascending=False)
df2 = df1[df1.Count >= 100]
df4 = df.groupby("UserId", as_index=False).mean()
```

Screen 6.9. Grouping by user-id and calculating mean

The summaries of each product is combined with the product-id's and the clean up process is performed on the summary.

```
combine_summary = df2.groupby("UserId")["Summary"].apply(list)
combine_summary = pd.DataFrame(combine_summary)
combine_summary.to_csv("combine_summary.csv")
df3 = pd.read_csv("combine_summary.csv")
df3 = pd.merge(df3, df4, on="UserId", how='inner')
df3 = df3[['UserId','Summary','Score']]
df3.to_csv("df3.csv")
cleanup_re = re.compile('[^a-z]+')
def cleanup(sentence):
    sentence = sentence.lower()
    sentence = cleanup_re.sub(' ', sentence).strip()
    sentence = " ".join(nltk.word_tokenize(sentence))
    return sentence
df3["Summary_Clean"] = df3["Summary"].apply(cleanup)
```

Screen 6.10. Combining and clean-up of summary for UBCF

Now the duplicates from the dataframe are removed on the basis of score except the last tuple and then reset the index.

```
df3 = df3.drop_duplicates(['Score'], keep='last')
df3 = df3.reset_index()
```

Screen 6.11. Removing duplicates and reset the index for UBCF

From the summary obtained by the above step 100 features of the were extracted are transformed into vectors and a dataframe is created with the indices similar to summary_clean.

```
docs = df3["Summary_Clean"]
vect = CountVectorizer(max_features = 100, stop_words='english')
X = vect.fit_transform(docs)
df5 = DataFrame(X.A, columns=vect.get_feature_names())
```

Figure 6.12. Extracting features and vector transformation for UBCF

To create training and test data the dataset obtained by the above process is divided accordingly.

```
X = np.array(df5)
tpercent = 0.95
tsize = int(np.floor(tpercent * len(df5)))
df5_train = X[:tsize]
df5_test = X[tsize:]
lentrain = len(df5_train)
lentest = len(df5_test)
```

Figure 6.13. Division of training and test data for UBCF

Then a dataframe is created by removing the duplicates in the summary and reset the index.

```
kkk  = df.drop_duplicates(['Summary'], keep='last')
kkk = kkk.reset_index()
```

Figure 6.14. Removing duplicates in summary and reset the index

Next a nearest neighbor object has to be instantiated, and call it nbrs Then fit it to dataset X. To find the k-neighbors of each point in object X, call the kneighbors() function on object X and “Ball_Tree” algorithm is used to find three nearest neighbours.

```
nbrs = NearestNeighbors(n_neighbors=3, algorithm='ball_tree').fit(df5_train)
distances, indices = nbrs.kneighbors(df5_train)
```

Figure 6.15. Instantiating object for UBCF

After finding the related product, the user-id of the product and the user-id's in the dataset "kkk" is verified and if they matches then the score is verified. If the score is equal to five then the product is recommended. In this way the products for the users in test dataset are found and then the accuracy, precision, recall, F1 score, support are calculated.

6.3 Model Evaluation

The dimensions of the dataset can be known by using the command shape. The output of this command is the total number of rows and columns present in the dataset. The dimensions of the dataset is

```
(568454, 10)
```

Output of Product-based filtering

The total number of tuples in the "Reviews.csv" dataset after grouping the tuples based on product-id and removing duplicates of the summary is 361. In that 324 tuples are used in the training dataset and 37 tuples are used as test dataset. The first three similar product-id's for the product in the dataset are displayed along with the average mean of the score for 37 products that are in the test dataset.

The summary after the clean-up process will be displayed and the index ranges from 0 to 360.

```
0     the only treat my dog will eat great for diabe...
1     very good dogs love these great for travel but...
2     blueray great movie blue ray okay beetlefun fa...
3     great movie but not given deluxe treatment my ...
4     it worked finally a trap that works it works b...
5     great movie one of the best movies ever fabulo...
6     the new favorite lasts we all love frank dog l...
7     formula changes causing upset great dog food v...
8     best canned wet food out there my cat loves it...
9     star rating best cat food in the world good he...
10    incredible at any price the best coffee i ve t...
11    easter will never be the same what a great ide...
12    good night sleep soothing great tea works econ...
13    post workout treats best bar get in my belly l...
14    deee lishhh tastes like burnt wood senseo pods...
15    decent coffee great price the folgers and mill...
16    not even close suggested by dr oz served its p...
17    yummy taste great tea reasonable price tastes ...
18    a very fine tea sage tea helping with hot flas...
```

Screen 6.16. Summary after clean-up process

```
Name: Summary_Clean, Length: 361, dtype: object
Based on product reviews, for  B007TGDXMU  and this average Score is  4.517006802721088
The first similar product is  B005ZBZLPI  and this average Score is  4.0
The second similar product is  B000SDKDM4  and this average Score is  4.08252427184466
The Third similar product is  B001E50THY  and this average Score is  4.044642857142857
-----------------------------------------------------------
Based on product reviews, for  B007TGDXNO  and this average Score is  4.3478260869565215
The first similar product is  B007TGDXMK  and this average Score is  4.143540669856459
The second similar product is  B00451WLYI  and this average Score is  4.517647058823529
The Third similar product is  B0001ES9F8  and this average Score is  4.302083333333333
-----------------------------------------------------------
Based on product reviews, for  B007TJGY5K  and this average Score is  4.119496855345912
The first similar product is  B007L3NVKU  and this average Score is  4.355072463768116
The second similar product is  B0027Z8VES  and this average Score is  4.341269841269841
The Third similar product is  B005HUVI40  and this average Score is  3.9917355371900825
-----------------------------------------------------------
Based on product reviews, for  B007TJGZ0Y  and this average Score is  4.384615384615385
The first similar product is  B002QGK2V8  and this average Score is  3.4484848484848483
The second similar product is  B0000V8IOE  and this average Score is  3.9274193548387095
The Third similar product is  B005HUVI40  and this average Score is  3.9917355371900825
```

Screen 6.17. Output of product-based filtering

```
                precision    recall  f1-score   support

           CFA       0.90      0.70      0.79        27

      accuracy                           0.73        37
     macro avg       0.70      0.75      0.70        37
  weighted avg       0.80      0.73      0.74        37

Accuracy is 0.7297297297297297
```

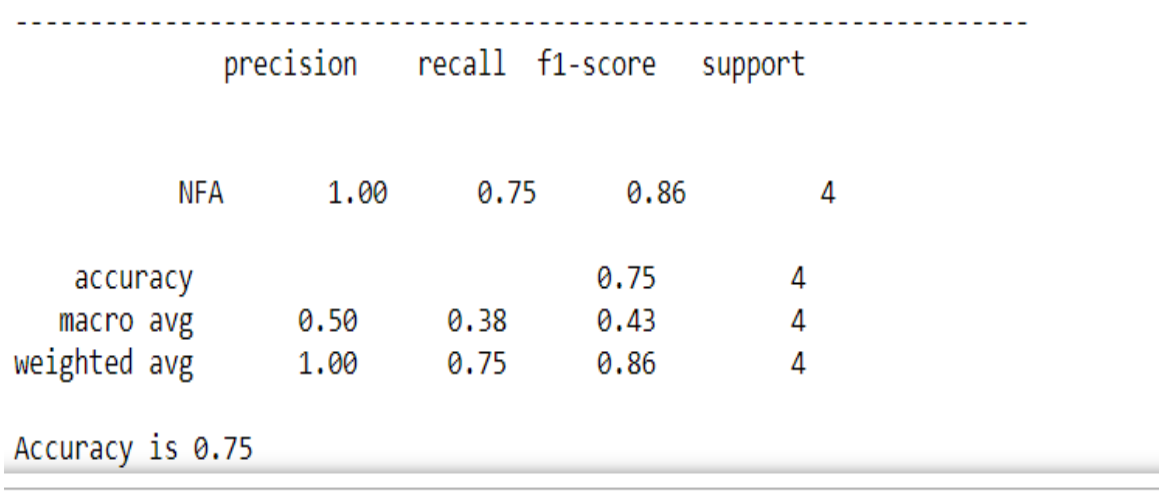
Screen 6.18. Accuracy of product-based filtering

Output of User-based filtering

The total number of tuples in the "Reviews.csv" dataset after grouping the tuples based on user-id and removing duplicates of the summary is 65. In that 48 tuples are used in the training dataset and 17 tuples are used as test dataset. The products for the user are recommended based on the similar user's purchase history for the 17 users in the test dataset. The product-id's of the products are displayed that can be recommended to the particular user having the user-id in the test dataset.

```
(568454, 10)
Based on  reviews, for user is  AY12DBB0U420B
The first similar user is  A3FKGKUCI3DG9U .
He/She likes following products
B007K449CE
B000EPUPSS
B000F4DKAS
B001OCBT3U
B000EMM9WG
B0014X5O1C
B004158VLU
B0018SMUVA
B004BKLHOS
B000F4J76E
B000YCJRIU
B008RWUHA6
B000MPQ4Q2
B000ETVRQS
B001E6IUMY
B000EMK4CS
```

Screen 6.19. Output of user-based filtering



	precision	recall	f1-score	support
NFA	1.00	0.75	0.86	4
accuracy			0.75	4
macro avg	0.50	0.38	0.43	4
weighted avg	1.00	0.75	0.86	4

Accuracy is 0.75

Screen 6.20. Accuracy of user-based filtering

6.4 Dimensionality Reduction

Dimensionality reduction, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data. When dealing with high dimensional data, it is often useful to reduce the dimensionality by projecting the data to a lower dimensional subspace which captures the “essence” of the data. This is called dimensionality reduction. A dataset contains a huge number of input features in various cases, which makes the predictive modeling task more complicated. Because it is very difficult to visualize or make predictions for the training dataset with a high number of features, for such cases, dimensionality reduction techniques are required to use.

The dataset “Reviews.csv” contains 10 attributes called Id, Product-id, user-id, profile name, helpfulness numerator, helpfulness denominator, score, time, summary, text.

In “Reviews.csv” dataset only few attributes are used in both product based and user based modules. For product-based product-id, score, summary and calculated value of count is used and the rest are discarded by renaming user-id as count and score_x as score and summary_x as summary and a dataframe is created with is attributes.

```
count = df.groupby("ProductId", as_index=False).count()
df1["Count"] = df1["UserId_y"]
df1["Score"] = df1["Score_x"]
df1["Summary"] = df1["Summary_x"]
df1 = pd.merge(df, count, how='right', on=['ProductId'])
df1 = df1[['ProductId','Summary','Score',"Count"]]
```

Figure 6.21. Dimensionality reduction in product-based filtering

For user-based user-id, score, summary and calculated value of count is used and the rest are discarded by renaming product-id as count and score_x as score and summary_x as summary and a dataframe is created with is attributes.

```
count = df.groupby("UserId", as_index=False).count()
df1["Count"] = df1["ProductId_y"]
df1["Score"] = df1["Score_x"]
df1["Summary"] = df1["Summary_x"]
df1 = pd.merge(df, count, how='right', on=['UserId'])
df1 = df1[['UserId','Summary','Score',"Count"]]
```

Figure 6.22. Dimensionality reduction in user-based filtering

CONCLUSION

Recommendation systems have been an important in E-commerce on the web for the customer to suggest items what they would be interested. With the increasing number of users and items, recommendation systems encounter the main shortcoming: data sparsity and data scalability problems, which bring out the poor quality of prediction and the inefficient time consuming.

In this report the collaborative recommendation systems were discussed and product-based filtering and user-based filtering were implemented using K-Nearest neighbor algorithm. In product-based filtering for each product the first 3 similar products are identified and accuracy is calculated. In user based the products are recommended based on the similarity between the users and the accuracy is calculated. The products and users are uniquely identified by product-id and user-id. The accuracy of user-based filtering is more when compared to product-based filtering.

BIBLOGRAPHY

Journals

[1] Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowl Based Syst 46:109–132.

[2] Y. Guo, M. Huang, T. Lou (2015) A Collaborative Filtering Algorithm of Selecting Neighbors Based on User Profiles and Target Item (Accessed 22-02-2016)

[3] Thorat, P. B., Goudar, R. M., & Barve, S. (2015). Survey on collaborative filtering, content based filtering and hybrid recommendation system. International Journal of Computer Applications, 110(4), 31-36.

[4] Peng Yu (2015) Collaborative Filtering Recommendation Algorithm Based on Both User and Item(Accessed 21-02-2017)

[5] Gong, Songjie. 2010. "A Collaborative Filtering Recommendation Algorithm Based On User Clustering And Item Clustering." JSW 5 (7). doi:10.4304/jsw.5.7.745-752.

Conference papers

[6] Xingyuan Li.2011 "Collaborative Filtering Recommendation Algorithm Based on Cluster", International Conference on Computer Science and network Technology(ICCSNT), IEEE, 4: 2682-2685.

[7] Schafer, J. Ben, Joseph Konstan, and John Riedl. 1999. "Recommender Systems In E-Commerce." In 1St ACM Conference On Electronic Commerce, 158-166.

[8] Resnick, Paul, Iacovou, Neophytos, Suchak, Mitesh, Bergstrom, Peter, Riedl, John.1994. " GroupLens: an open architecture for collaborative filtering of netnews." CSCW conference, ACM (1994).