Koneru Lakshmaiah Education Foundation

(Deemed to be University)

RECOMMENDER SYSTEM WITH CONTENT-BASED FILTERING METHOD

A Major Project Report

Submitted in the partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

In

Department of Computer Science

By

180030154 – A. SAI KOUSHIK

180030659 – G. SRI CHARAN

Under the Esteemed Guidance of

Dr. P. V. R. D. PRASADA RAO

Professor

Department of Computer Science Engineering

K LE F, Green Fields

Vaddeswaram-522502, Guntur (Dist), Andhra Pradesh, India.

November 2021

DECLARATION

The Project Report entitled “RECOMMENDER SYSTEM USING CONTENT-BASED FILTERING” is a record of bonafide work of A. Sai Koushik (180030154), G. Sri Charan (180030659), submitted in partial fulfilment for the award of B. Tech in Computer Science to the K L University. The results embodied in this report have not been copied from any other Departments/ University/ Institute.

Name	ID NO.
A. Sai Koushik	180030154
G. Sri Charan	180030659

CERTIFICATE

This is to certify that the Report entitled " RECOMMENDER SYSTEM USING CONTENT-BASED FILTERING " is being submitted by A. Sai Koushik (180030154), G. Sri Charan (180030659) submitted in partial fulfilment for the award of B. Tech in Computer Science to the K L University is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other Departments/ University/ Institute.

Signature of the Co-Supervisor **Signature of the Supervisor**

Signature of the HOD **Signature of the External Examiner**

ACKNOWLEDGEMENT

My sincere thanks to **Dr. P. V. R. D. Prasada Rao** in the minor project for their outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **DR. HARI KIRAN VEGE**., Head of the Department for Computer Science and Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project.

We would like to place on record the deep sense of gratitude to the honourable Vice Chancellor, K L University for providing the necessary facilities to carry the concluded project.

Lastly, we thank all Teaching and Non-Teaching Staff of our department and especially my classmates and my friends for their support in the completion of our project.

Project Associates….

Atluri Sai Koushik 180030154

Gunisetty Sri Charan 180030659

TABLE OF CONTENTS

ABSTRACT

Finding data on an enormous site can be tedious and confounding. Recommender frameworks can help clients in finding data by making redid proposals. Recommender frameworks, otherwise called proposal frameworks, are a sort of data sifting framework that predicts the client's 'appraisal' or 'inclination' for a given thing. Content-based proposal frameworks endeavor to recommend things that are practically identical to those that a client has recently delighted in. For sure, a substance-based recommender's center technique involves looking at the properties of a client profile, which stores inclinations and interests, with the qualities of a substance object (thing), to propose new fascinating things to the client. This paper gives an outline of the substance-based separating strategy, which is utilized in E-trade locales and Online Social Networks to reap information from different clients and afterward use that information to further develop Recommender frameworks.

We show that our technique outflanks a current social-separating strategy in the space of film proposals on a dataset of in excess of 100236 film appraisals gathered from a local area of more than 600 clients.

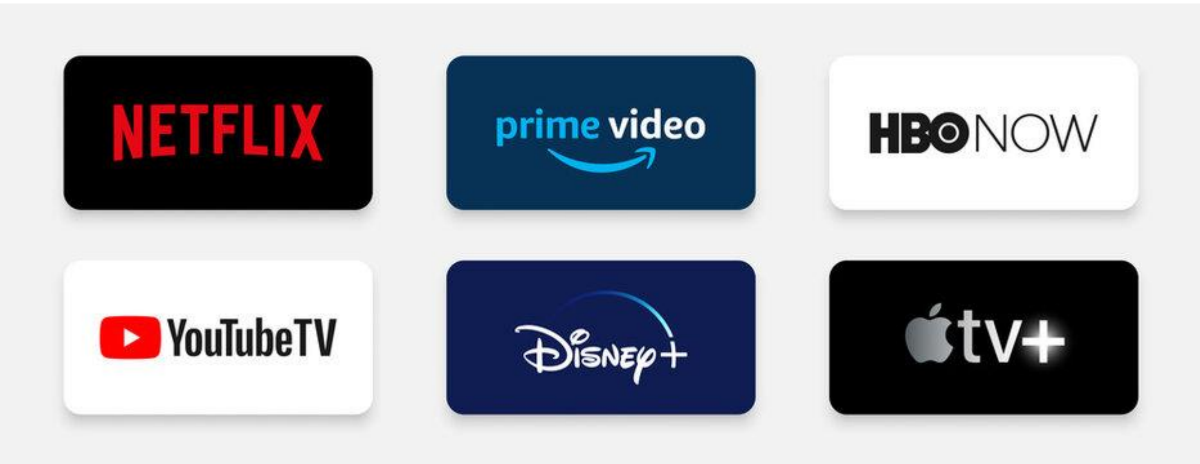
Keywords— Recommender Systems, Content-based filtering, Movie Recommendation, Cosine similarity.

1. INTRODUCTION

Recommender System

It is to produce significant proposals to an assortment of clients for things or items that may intrigue them. Thoughts for books on Amazon, or movies on Netflix, are veritable cases of the action of industry-strength recommender structures. The plan of such suggestion motors relies upon the area and the specific qualities of the information accessible. For instance, film watchers on Netflix every now and again give evaluations on a size of 1 (loathed) to 5 (loved). Such an information source records the quality of cooperation's among clients and things. Also, the framework might approach client explicit, and thing explicit profile ascribes like socioeconomics and item portrayals separately. Recommender frameworks vary in the manner they examine these information sources to foster thoughts of proclivity among clients and things which can be utilized to recognize very much coordinated with sets.

The objective of a Recommender System is to create significant proposals to an assortment of clients for things or items that may intrigue them. Ideas for books on Amazon, or motion pictures on Netflix, are certifiable instances of the activity of industry-strength recommender frameworks. The plan of such suggestion motors relies upon the area and the specific attributes of the information accessible. Recommender frameworks were created to assist with shutting the hole between data assortment and investigation by sifting all of the accessible data to introduce what is generally significant to the client.



Today, when the web has turned into a basic piece of human existence, individuals are experiencing issues choosing what to purchase because of the immense range of choices. There is just too much information available on the internet while looking for everything from a model to decent investing possibilities. Companies have implemented recommendation systems to assist users in dealing with this information deluge. Despite the fact that there has been research around here of suggestion frameworks for quite a while, interest stays high because of the wealth of common-sense applications and the issue-rich area. Recommendations are a common occurrence in everyday life. Recommendation techniques are most recognized for their application on e-commerce websites, where they provide a list of recommended things based on information about a customer's preferences. Many internet applications analyse consumer behaviour in terms of products purchased, items watched, and user ratings in order to recommend items to their users depending on their interests.

E-commerce stores utilise recommendation algorithms to personalise the online store for each consumer in order to entice them by offering things that match their interests regularly. Recommender Systems have evolved into a valuable tool for providing individualised recommendations on things such as clothing, books, movies, music, and shoes to users. Table 1 shows some famous destinations which are presently utilizing proposal framework for various reason

Table 1: Popular sites using recommender systems

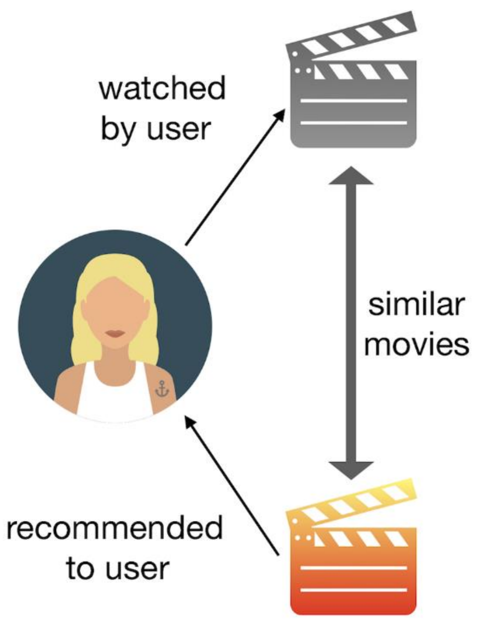
Site	Recommendation Category
Netflix, prime, Hulu	Movies & Series
Amazon	Books & products
Facebook	Friends & Shopping
LinkedIn	Jobs
Spotify, YT Music	Songs

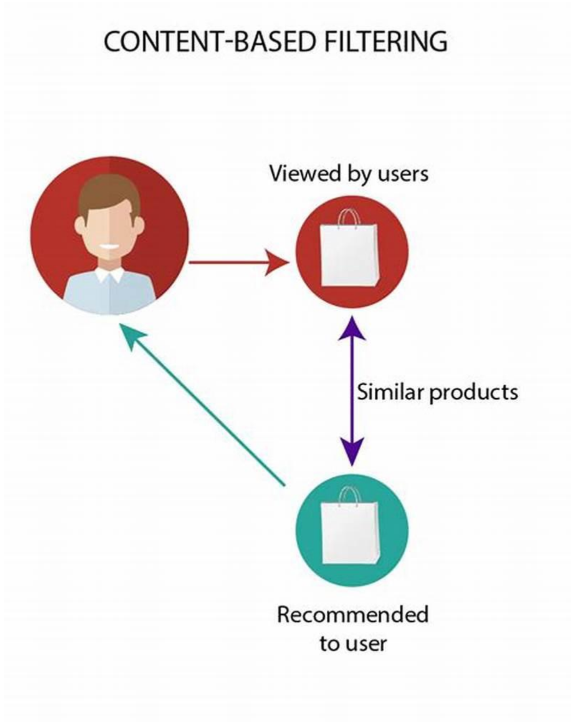
Content-Based Filtering

Content-based filtering, here and there alluded to as intellectual separating, recommends things dependent on an examination of the things content and a client profile. Each item's content is represented by a set of descriptors or terms, which are often words found in a text. The user profile is described in the same words and is created by assessing the content of items that the user has viewed. The user will be recommended things that are primarily connected to the positively rated items. In order to create useful suggestions, Content Based Filtering employs a variety of models to discover similarities between documents. Since different clients' profiles don't impact the proposal, the substance-based sifting procedure doesn't need them. Furthermore, if the user profile changes, the Content Based Filtering approach may still alter its recommendations in a very short amount of time. In the content-based sifting method, the thing depiction and client’s profile has a ton of significance. It follows the following steps: -For recommendation attributes of the items are to be defined -The active user’s preferences and the attributes of the items are compared. Things are recommended according to the interest of the user.

Content-based isolating, moreover, insinuated as scholarly filtering, proposes things reliant upon a relationship between the substance of the things and a customer profile. The substance of everything is addressed as a bunch of descriptors or terms, normally the words that happen in a report. The client profile is addressed with similar terms and developed by breaking down the substance of things which have been seen by the client. Things that are by and large related to the determinedly assessed things are endorsed to the customer. CBF utilizes various kinds of models to track down closeness between records to produce significant suggestions. Content-based separating strategy needn't bother with the profile of different clients since they don't impact suggestion. Additionally, if the client profile changes, CBF strategy can possibly change its suggestions inside an extremely brief timeframe.

For example, in the event that a customer partakes in a page with the words convenient, pen drive, and RAM, the CBF will endorse pages related to the equipment world. Thing depiction and a profile of the customers bearing accept a huge part in Content-based isolating. Content-based filtering estimations endeavour to recommend things reliant upon equivalence count. The best-coordinating with things is suggested by contrasting different up-and-comer things and things recently evaluated by the client. A substance-based recommender works with information that the client gives, either unequivocally (evaluating) or certainly (tapping on a connection). In view of that information, a client profile is created, which is then used to make ideas to the client. As the client gives more information sources or makes moves on the proposals, the motor turns out to be increasingly precise.





Content put together separating strategy is based with respect to the semantic search that is the data retrieval. The other name for content based separating strategy is the intellectual separating. This strategy recommends things based on client's profile and the client's thing profile. The customer's profile is made when the customer starts the system. It gathers the interest of the clients and suggest the things after breaking down the highlights of the things and the clients. The suggested things are indistinguishable from the things that were preferred by the client prior or beforehand and they additionally match the properties of the client. This procedure functions admirably just when the ascribes are introduced in a reasonable and appropriate manner. In content based separating strategy the thing depiction what's more, client's profile has a great deal of significance.

It follows the accompanying advances:

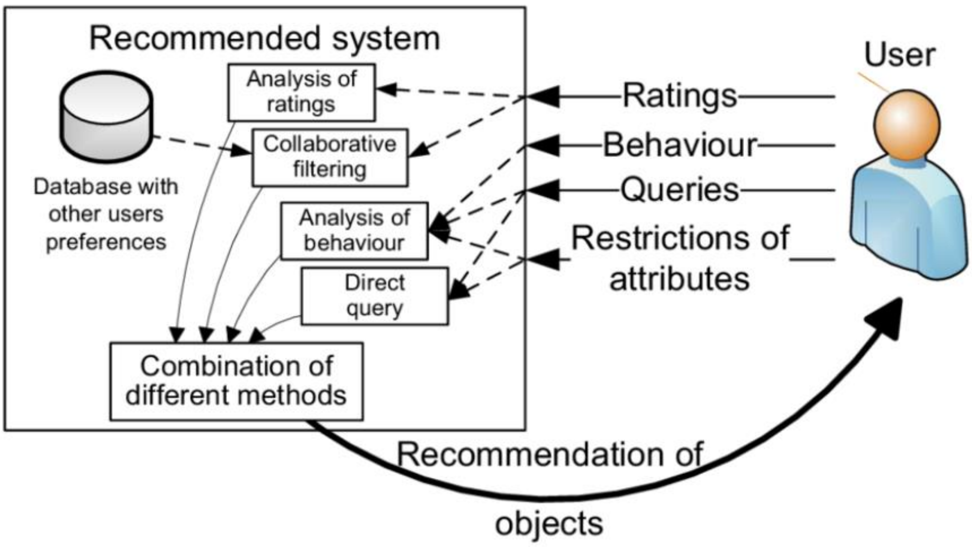
- For proposal ascribes of the substance are characterized

- The dynamic client's inclinations and the qualities of the content are analysed.

- Things are recommended in the interest of the customer.

Movie Recommendation

As recently expressed, a suggested framework is given as info a bunch of appraisals of individual relics for a particular client in the social separating approach. On account of film proposals, for instance, this information would be a rundown of motion pictures that the client had seen, each with a number rating. The recommendation system's output is a new group of artefacts that the user hasn't yet scored but that the recommendation system believes the user will like. This problem would be solved by social-filtering systems focused entirely on each user's movie ratings and computing a function from these appraisals can give a client a rating for a film that others have evaluated yet the client has not. Rather than a binary label, these methods have generally produced movie ratings. They come up with ratings for unseen objects by comparing people's preferences for the rated items. Individual users of a system make similarity judgments, which are computed using several statistical methodologies. Recommender, for instance, ascertains a more modest gathering of reference clients known as the recommenders for each user. These recommenders are individuals of the local area who are the most like the client. The appraisals of these recommenders are then used to figure the evaluations of new motion pictures utilizing relapse strategies. The user is usually given with a rank-ordered list of recommended movies in this social recommendation strategy. On the opposite side, content-based proposal calculations would just think about non-evaluations information. They would take a portrayal of each cherished and detested film for every client and become familiar with a strategy that would take a new film depiction and foresee whether the client might want or aversion it. A different suggestion technique would be utilised for each user. The reason for our examination is to make a suggestion framework that thinks about the two evaluations and content data. By characterizing the issue as one of grouping rather than ancient rarity appraisal, we wander from the commonplace social-separating way to deal with the suggestion. On the other hand, we differ from content-based filtering approaches in that we will employ social information in the inductive learning process, in the form of other users' evaluations. We'll figure the film proposal issue as a learning issue, explicitly the test of learning a capacity that takes a client and a film as sources of info and yields a mark showing whether the film is loved (thus recommended) or disdain.



Advantages of Content-Based Filtering:

- Content based filtering strategy gives the client independency based on the elite evaluations that are utilized to assemble the client profile.
- In this method the client becomes acquainted with how the recommender framework processes that is there is part of straightforwardness.
- CBF additionally suggests those things which have not been utilized by any individual and this is by implication good for the current client.
- A significant benefit of content-based sifting is that clients can get understanding into the inspiration why things are thought of as applicable to them on the grounds that the substance of everything is known from its portrayal.
- Content-based filters are less affected by problems of collaborative filtering systems such as "cold start" and sparsity.

Disadvantages of Content-Based Filtering:

- Creating the attributes of the things are troublesome in some specific field.
- Content based filtering suffers from the overspecialization problem as it recommends items that are all same type. The user won't get the items beyond the predefined boundary which leads to poor recommendation.
- In this method we cannot check whether the suggested is right or not on the grounds that in CBF the client's input isn't gathered as the clients don't give the evaluations for the things.
- CBF just uses the measurement information of the individual, yet the connection information of the individual isn't utilized.
- Content-based separating has a significant weakness, for instance, the way that it centers around catchphrase closeness. This methodology is unequipped for catching more complicated connections at a more profound semantic level, in light of various kinds of traits related to organized objects of the text. Therefore, numerous things are missed, and numerous superfluous things are recovered.

Limitations of Recommender Systems:

Data sparsity

Obviously, the utilization of recommender frameworks is consistently expanding. Subsequently, numerous business huge datasets are utilized in recommender frameworks. Therefore, the separating client thing grid could be very enormous. Large and sparse, and as a result of their performance, it’s possible that the recommendation process will deteriorate. The weather was freezing. The information sparsity is the foundation of the issue. In the Recommendation of a community, the sifting approach thing depends on past client choices, so new clients should rate an adequate number of products to qualify. Enabling the system to accurately capture their choices as a result, genuine recommendations are possible.

Scalability

As the number of clients and merchandise develops, customary CF calculations will confront adaptability issues. Consider a ten-million-customer O(M) and-million-item O(N) scenario; the complexity of the algorithm is 'n,' which is already excessive. As a recommender system, it is critical in E-commerce applications because the system must reply to the user's request instantly and offer recommendations regardless of the user's rating history or transactions, which necessitates a larger level of scalability. Twitter is a gigantic web company that utilizes bunches of machines to scale the ideas of their a great many clients.

Vulnerability to attacks

Security is a pivotal worry with each framework that is put on the web. Since recommender frameworks have such a significant impact on web-based business applications, they are probably going to be focuses of pernicious assaults pointed toward advancing or restraining explicit things. This is quite possibly the main challenge that the recommender framework's designer faces.

Diversity

Recommender frameworks are expected to expand assortment since they assist us with finding recently dispatched items. Some algorithms may accidentally do the reverse. Here recommender system recommends

Well-known and exceptionally appraised things which are valued by a specific client. This prompted lower precision in the suggestion interaction. To beat this issue there is a need to foster new mixture approaches which will upgrade the proficiency of the suggestion cycle.

2. LITERATURE SURVEY

S.NO	Paper Name	Author	Published By	Year	Methodology	Methodologies common to our project.
1.	A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System.	Geetha G Safa M Fancy C Saranya D	IOP- Institute of Physics Publishing	2018	Hybrid Filtering	Content Based Filtering
2.	Recommender Systems	Prem Melville, Vikas Sindhwani	IBM T.J. Watson Research Center	2016	Introduction To recommendation systems.	None
3.	Recommender Systems: Types of Filtering Techniques	L. Chameikho Iateilang Ryngksai	International Journal of Engineering Research & Technology	2014	Types of Filtering Techniques	Content Based Filtering
4.	Recommendation as Classification: Using Social and Content-Based Information in Recommendation:	Chumki Basu Haym Hirsh William Cohen	AAAI - Association for the Advancement of Artificial Intelligence.	2010	Social and Content-based Information Filtering.	Content-Based Information Filtering.
5.	A Content Based Movie Recommender System Empowered by	HİLAL KARAMAN	Applied Sciences of Middle East Technical Univeristy at	2010	Content-Based Filtering And	Content-Based Filtering.

	Collaborative Prediction:		The Graduate School of Natural.		Collaborative Based Filtering.	
6.	An Ontology-Content-Based Filtering Method	Peretz Shoval, Veronica Maidel, Bracha Shapira	International Journal "Information Theories & Applications"	2008	Ontology Content-Based Filtering.	None

A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System:

MOVREC is a movie recommendation system presented by D.K. Yadav et al. based on collaborative filtering approach. Shared separating utilizes data given by the client. That data is broke down, and a film is prescribed to the clients which are organized with the film with the most elevated rating first. The framework additionally has an arrangement for the client to choose ascribes on which he needs the film to be suggested. Luis M Capos et al. have investigated two customary recommender frameworks i.e., content-based sifting and shared separating. As both have their own downsides, he proposed another framework that is a mix of a Bayesian organization and community oriented sifting. The proposed framework is advanced for the given issue and gives likelihood disseminations to make helpful inductions. A half breed framework has been introduced by Harpreet Kaur et al. The framework utilizes a blend of content just as a synergistic sifting calculation. The setting of the motion pictures is likewise thought of while suggesting. The client relationship, just as the client thing relationship, assumes a part in the proposal. The client explicit data or thing explicit data is clubbed to shape a bunch by Utkarsh Gupta et al. utilizing chameleon. This is a productive strategy dependent on progressive grouping for the recommender frameworks. To anticipate the rating of a thing casting a ballot system is utilized. The proposed framework has lower blunders and has a better grouping of comparative things. UrszulaKuelewska et al. proposed grouping to manage recommender frameworks. Two strategies of computing group delegates were introduced and assessed. Centroid-based arrangement and memory-based collective separating techniques were utilized as a reason for the contrasting effects of the proposed two strategies. The outcome was a huge expansion in the precision of the generated proposals when contrasted with just the centroid-based technique. Costin-Gabriel Chiru et al. proposed Movie Recommender, a framework that utilizes the data known about the client to give film proposals. This framework endeavors to tackle the issue of unique suggestions that come about because of overlooking the information explicit to the client. The psychological profile of the client, their watching history, and the information including film scores from different sites is gathered. They depend on total closeness computation. The system is a half-breed model which utilizes both substance-based separating and community filtering. To foresee the trouble level of each case for every student Hongli Lin et al. proposed a method called content helped community separating (CBCF). The calculation is partitioned into two phases, the first being the substance-based separation that further develops the current student case appraisals information and the second being communitarian sifting that gives the last forecasts.

Recommender Systems by Prem Melville and Vikas Sindhwani:

In this paper, they examined the particular suggestion frameworks like Collaborative, content-based, and Hybrid filtering. Pure Collaborative Filtering recommenders just use the client rating lattice, either straightforwardly, or to prompt a cooperative model. These methodologies treat all clients and things as nuclear units, where forecasts are made regardless of the particulars of individual clients or things. Be that as it may, one can make a superior customized proposal by find out about a client, like segment data, or about a thing, like the chief and class of a film. For example, given film type data, and realizing that a client preferred Star Wars and Blade Runner, one might deduce an inclination for Science Fiction and could subsequently suggest Twelve Monkeys. Content-based recommenders allude to such methodologies, that give proposals by contrasting portrayals of content depicting a thing to portrayals of content that interest the client. These methodologies are occasionally likewise alluded to as content-based sifting. Much examination in this space has zeroed in on suggesting things with related literary substance, for example, website pages, books, and motion pictures, where the site pages themselves or related substance like depictions and client surveys are accessible. Thusly, a few methodologies have regarded this issue as an Information Retrieval (IR) the errand, where the substance related with the clients inclinations is treated as a question, and the unrated archives are scored with significance/likeness to this question. In News Weeder, reports in each evaluating classification are changed over into the-IDF word vectors and afterward found the middle value of to get a model vector of every class for a client. To characterize another archive, it is contrasted and every model vector and given an anticipated rating dependent on the cosine comparability to every class. An option in contrast to IR draws near, is to treat suggesting as an order task, where every model addresses the substance of a thing, and a client's past evaluations are utilized as marks for these models In the space of the book suggesting, Mooney et al. use text from fields like the title, maker, abstracts, surveys, furthermore, subject terms, to set up a multinomial nave Bayes classifier. Appraisals on a scale of 1 to k can be straightforwardly planned to k classes, or on the other hand, the numeric rating can be used to weigh the arrangement model in a probabilistic twofold gathering setting. Other order calculations have likewise been utilized for simply content-based suggesting, including a k-closest neighbour, choice trees, and neural organizations.

Recommender Systems: Types of Filtering Techniques:

Recommender frameworks can make upper hands for organizations that carry out them and consumer loyalty can be accomplished alongside client devotion. Organizations are bound to be constrained out of the market assuming it doesn't carry out RS. In this paper we talk about three distinct sorts of RS however it is hard to express that a specific RS is better than the other as very basic frameworks can likewise be less expensive to carry out even though give more slow exactness. In any case, practically every RS will have issues when first carried out as there is insufficient information accessible on clients what's more, things. Further develop the precision when first carrying out the RS since awful proposals to clients can diminish the impact RS have on deals. Other expected issues in RS can be brought about by malevolent clients thus the precision of the RS could be impacted when huge amounts of phony profiles are embedded. This may conceivably lead to terrible suggestions and subsequently potentially add to a decrease of the impact RS have on deals.

Recommendation as Classification: Using Social and Content-Based Information in Recommendation:

The data from this work is to foster a way to deal with proposals that can take advantage of the two evaluations and content data. We withdraw from the conventional social-separating way to deal with a proposal by outlining the issue as one of grouping, rather than curio rating. Then again, we contrast from content-based sifting techniques in that friendly data, as different clients appraisals, will be utilized in the inductive learning process. Specifically, we will formalize the film suggestion issue as a learning issue - explicitly, the issue of learning a capacity that takes as its feedback a client and a film and creates as yield a mark demonstrating whether the film would be preferred (and hence suggested) or loathed:

$$f(\langle user, movie \rangle) \rightarrow \{liked, disliked\}$$

As an issue in characterization, we additionally are keen on foreseeing whether a film is loved or detested, not a careful rating. Our yield is additionally not an arranged rundown of motion pictures, but rather a bunch of films that we anticipate will be loved by the client. Above all, we are presently ready, to sum up, our contributions to the issue to other data portraying the two clients and films. The data we have accessible for this interaction is an assortment of client/film evaluations (on a size of 1-10), and certain extra data concerning each movie.1 To introduce the outcomes resources of motion pictures anticipated to be loved or hated by a client we figure a rating limit for every client with the end goal that 1/4 of the multitude of clients appraisals surpass and the leftover 3/4 don't, and we return as suggested any film whose anticipated rating is over the preparation information put together edge with respect to films.

A Content Based Movie Recommendation System Empowered by Collaborative Missing Data Prediction:

This proposal study addresses an online film suggestion framework which is named as ReMovender. As explained throughout this work, ReMovender is composed of many different approaches and algorithms, each of which help to increase the capacity and success of the system. As the first improvement, demographic data is embedded into the user-based similarity part. With the addition of demographic data, the system is aimed to make more powerful recommendations. However, the distance measure of some of the demographic features could have been formulated in a different way so that the demographic similarity between two users is calculated more accurately. For the gender orientation highlight, the distance between the gender of two users is taken as 1 assuming the clients have similar gender and 0 in any case. This formula seems to be reasonable since this cannot be done in a different way. Similarly, the distance between the ages of two users is calculated with a formula which can be considered as a rational way of comparing ages. However, the distance formula between occupations and zip codes may be replaced with a better formula. In future work, the distance between postal divisions can be calculated by utilizing the geological distance between the comparing areas.. Besides, the occupations which are related with each other can be grouped together. By this way, dissimilar but related occupations will be assigned a distance value between 0 and 1 instead of 0. As for the content-based part, the number of features which are used to correlate movies can be increased in the future. As stated before, relying on the results of an existing study, features having consistent weight values are used in 93 ReMovender. In the future, as ReMovender grows and owns a greater number of users, these weight values can be altered and new features having their own weight values can be added. This can be done in different ways such as requesting feedback from users or analysing the ratings given by them. Some improvements can also be done in the evaluation part as a future work. In the evaluation part, the weight values assigned to movie features are assumed to be the values producing the best results. As a future work, this assumption can be supported by adding some numerical analysis and providing some graphical results.

An Ontology- Content-Based Filtering Method by Peretz Shoval, Veronica Maidel, Bracha Shapira:

In this research paper they presented a new content-based filtering method that uses ontology for representing user and item profiles, and for ranking items according to their relevancy in the electronic newspaper's domain. The technique is being executed in the e-Paper framework for customized electronic papers. The sifting technique thinks about the progressive distance, closeness, between ideas in the client's profile and ideas in the things' profile. The method can be enhanced in various aspects. One possible enhancement is to assign more importance to concepts co-occurring in items read in the past by the user. An item which includes co-occurring concepts might get a higher score than an item including the same concepts that did not co-occur in past read items. The added value of the incorporation this enhancement will be examined before being implemented in the method. Another possible enhancement of the method is to consider penalty scores for concepts appearing in an item but not in the user's profile. This idea, which was adopted from Savia et al. [1998], means that a concept in an item's profile which does not appear in the user's profile might be given a negative (penalizing) score. The contribution of such penalty to the quality of the filter can be determined in empirical experiments. The proposed separating strategy uses a 3-level progressive cosmology of News. It can be that as it may be summed up to different spaces with their particular ontologies, and it should not be limited to three levels. Moreover, the method can be enhanced to deal not just with a hierarchical but also with a network-based (DAG) ontology, where an idea might have many parent ideas, not just youngster ideas. One more conceivable expansion to the technique is to think about more sorts of relations between ideas, other than parent-youngster and grandparent-grandkid, e.g., twins of ideas. For instance, a client's profile might incorporate 'football' while a thing might incorporate 'ball'. These augmentations will be managed in the additional examination.

3. THEORETICAL ANALYSIS

Basic Requirements:

1. PC or Laptop
2. Jupyter Notebook (Anaconda)
3. Python (Programming Language)
4. Modules & Libraries in Python
5. Datasets (movies.csv & ratings.csv)

Python Scripting Language:

Python has become quite possibly the most famous programming language on the planet as of late. It's utilized in everything from AI to building sites and programming testing. It tends to be utilized by engineers and non-designers the same. Python, quite possibly the most famous programming language on the planet, has made everything from Netflix's suggestion calculation to the product that controls self-driving vehicles. Python is a universally useful language, which implies it's intended to be utilized in the scope of utilizations, including information science, programming and web advancement, computerization, and for the most part finishing stuff.

What is Python?

Python is a PC programming language frequently used to fabricate sites and programming, computerize errands, and direct information examination. Python is a universally useful language, which means it tends to be utilized to make a wide range of projects and isn't particular for a particular issue. This adaptability, alongside its novice kind disposition, has made it one of the most-utilized programming dialects today. A review led by industry examiner firm Red Monk observed that it was the most well-known programming language among engineers in 2020.

What is Python used for?

Python is normally utilized for creating sites and programming, task mechanization, information investigation, and information representation. Since its somewhat simple to learn, Python has been embraced by numerous non-developers like 27 bookkeepers and researchers, for a variety of everyday tasks, like organizing finances.

Installation Steps:

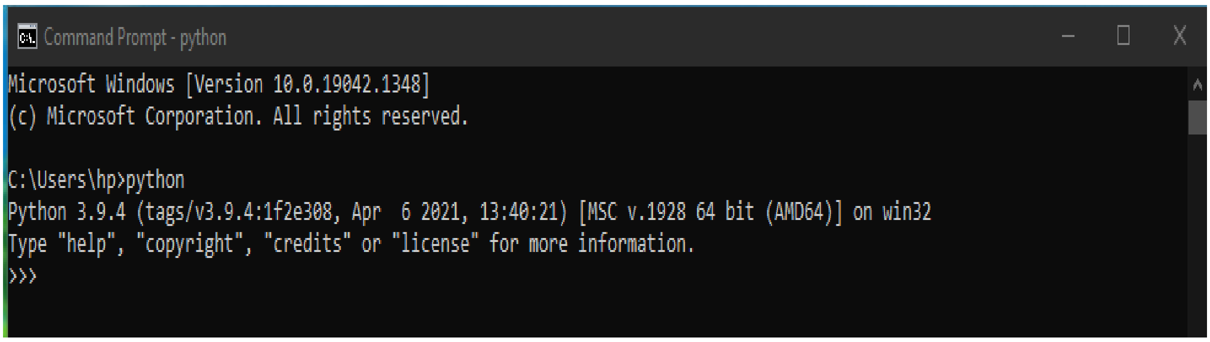
Step 1 – Select Version of Python to Install. Python has various versions available with differences between the syntax and working of different versions of the language. We need to choose the version which we want to use or need.

Step 2 –Download Python Executable Installer. On the internet browser, in the authority website of python, move to the Download for Windows area.

Step 3 – Run Executable Installer. We downloaded the Python 3.9.1 Windows 64-bit installer. Run the installer. Make a point to choose both at the base and afterward click Install New.

Step 4 – Verify Python is installed on Windows.
To guarantee in case Python is effectively introduced on your framework. Follow the given steps

- Open the command prompt.
- Type 'python' and press enter.
- The version of the python which you have installed will be displayed if the python is successfully installed on your windows.



Jupyter Notebook:

The Jupyter Notebook is an open-source web application that you can use to make and share records that contain live code, conditions, representations, and messages. Jupyter Notebook is kept up with by individuals at Project Jupyter.

Jupyter Notebooks are a side project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the center upheld programming dialects that it upholds: Julia, Python, and R. Jupyter ships with the IPython piece, which permits you to compose your projects in Python, however, there are as of now north of 100 different portions that you can likewise utilize.

Installation:

The best method of installing the Jupyter Notebooks is by the installation of the Anaconda package. The Jupyter Notebook and the Jupyter Lab comes pre-installed in the Anaconda package, and you don't have to install this on your own.

One of the prerequisites here is Python, either Python 3.3 or more noteworthy or Python 2.7. The overall proposal is that you utilize the Anaconda dissemination to introduce both Python and the scratchpad application. The upside of Anaconda is that you approach north of 720 bundles that can without much of a stretch be introduced with Anacondas conda, a bundle, reliance, and climate supervisor.

However, if you are not interested in installing the Anaconda package manager, and you want to simply install the Jupyter Notebook the Pythonic way, make sure you have downloaded the latest Python Version (which is 3.9 at the time of writing of this article), and then proceed to issue the following command in the command prompt or PowerShell.

```
pip install jupyter
```

With the installation of the Jupyter Notebook complete, you should be able to access it successfully. Simply type the jupyter note pad order in the order brief or PowerShell. This should launch your Jupyter Notebook in your selected web browser with a local hosting URL, which is labelled as http://localhost:8888.

Modules & Libraries used:

In Python Scripting Language, we have many libraries & modules to perform multiple and different operations. Some of these examples of these modules are: NumPy, Pandas, OpenCV, Me etc. The modules that we had used in our project are.

How to install modules in Python?

You can introduce modules or bundles with the Python bundle chief (pip). To introduce a module framework-wide, open a terminal and utilize the pip order. In the event that you type the code underneath it will introduce the module.

Command: pip install module-name.

That will install a Python module automatically. Generally, you do not install modules system wide, but use a virtual environment.

PANDAS & NUMPY:

Pandas is an open-source Python bundle that is most broadly utilized for information science/information examination and AI undertakings. It is based on top of another bundle named NumPy which offers help for multi-dimensional clusters. As quite possibly the most famous datum fighting bundles, Pandas functions admirably with numerous different information science modules inside the Python biological system and is regularly remembered for each Python circulation. Pandas simplify it to do a considerable lot of the tedious, dull errands related to working with information, including Data purifying, Data fill, standardization, measurable examination and representation, and informal assessment.

NumPy is a Python library used for working with arrays. In Python we have lists that serve the purpose of arrays, but they are slow to process. Other than its undeniable logical uses, NumPy can likewise be utilized as a proficient multi-dimensional compartment of conventional information. Subjective datatypes can be characterized utilizing NumPy which permits NumPy to consistently and quickly incorporate with a wide assortment of information bases. NumPy is a library for the Python programming language, adding support for enormous, multi-dimensional clusters and lattices, alongside a huge assortment of significant level numerical capacities to work on these exhibits.

DATASETS:

We will utilize a public informational index for film appraisals from Grouplens.org. It's a compressed document from which I have separated two records for this activity. The crude records are in GitHub as ratings.csv and movies.csv.

This information is a subset of a bigger film focal point informational collection that was utilized to fabricate a cross breed film proposal framework, conveying both substance and shared separating. movies.csv contains 9742 motion pictures from 600 clients, ratings.csv has 100836 film appraisals.

Dataset 1: movies.csv

This dataset comprises 9742 films and 3 ascribes which are:

i. Movie_id
ii. Title (including with year)
iii. Genre (of the movie)

Dataset 2: ratings.csv

Ratings dataset is a combination of 600 users and each user review on the different movies which sum up a total of 100836 ratings. This dataset has 4 columns:

i. User_id
ii. Movie_id
iii. Rating
iv. Timestamp

4. EXPERIMENTAL INVESTIGATIONS

Similarity Measures:

A similitude measure, otherwise called a closeness work, is a genuine esteemed capacity that evaluates the comparable rate between two things in insights and related fields. Although there is no universal definition of a similarity measure, they are frequently the inverse of distance metrics, taking enormous qualities for comparative items and either zero or a negative incentive for objects that are fundamentally divergent. With regards to information mining, a comparability measure is a distance with aspects addressing object highlights.When the distance between two attributes is little, they are quite similar, however when the distance is large, we will see a low degree of resemblance.

There are a few distinct kinds of comparability distance measurements. Some of the similarity measures include:

1)Cosine Similarity 2) Manhattan distance 3) Euclidean distance

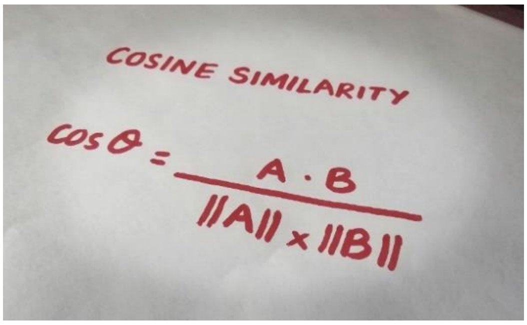
4)Minkowski distance 5) Jaccard Similarity

In this project, we will talk about Cosine similarity and the dot product, which we used to create a movie recommender system.

COSINE SIMILARITY:

Cosine comparability is a measurement used to gauge how comparable two things are. Numerically, it estimates the cosine of the point between two vectors projected in a multi-dimensional space. The output value ranges from **0–1**.

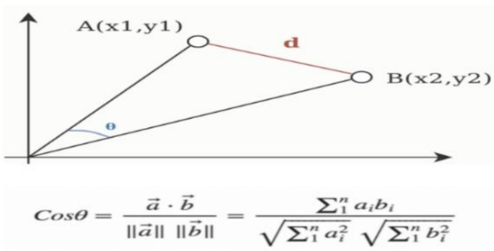
0 means no similarity, whereas 1 means that both the items are 100% similar.



The python Cosine Similarity or cosine portion processes likeness as the standardized speck result of info tests X and Y. We will utilize the sklearn cosine likeness to track down the cos for the two vectors in the counting network.

```
cosine_sim = cosine_similarity(count_matrix)
```

The cosine_sim framework is a NumPy cluster with determined cosine comparability between every film. As you can see in the image below, the cosine similarity of movie 0 with movie 0 is 1; they are 100% similar (as should be).



For instance, the cosine comparability between film 0 and film 1 is 0.105409 (a similar score between film 1 and film 0 request doesn't make any difference). Motion pictures 0 and 4 are more like one another (with a similitude score of 0.23094) than films 0 and 3 (score = 0.0377426). The inclining with 1s proposes what the case is, every film x is 100% such as itself.

$$similarity = cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}}$$

Dot Product:

The dot product is important when defining the similarity, as it is directly connected to it. The definition of similarity between two vectors **u** and **v** is, in fact, the ratio between their dot product and the product of their magnitudes. The dot product between two vectors is equal to the projection of one of them on the other. Therefore, the dot product between two identical vectors (i.e., with identical components) is equal to their squared module, while if the two are perpendicular (i.e., they do not share any directions), the dot product is zero. Generally, for *n*-dimensional vectors.

$$\langle x, y \rangle = \sum_{k=1}^{n} x_i y_i$$

Dot product as similarity:

Assume you have two vectors, x and y. To understand the geometric meaning of their dot product, consider that x can be split into two components: one parallel to y and the other orthogonal. The scaling factor r approaches the cosine of point between the things x and y since cosine is characterized by the proportion of the adjoining side and the hypotenuse.

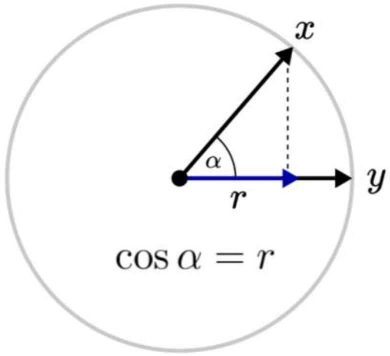


Fig: Dot Product

Approach/Workflow of Content-Based Filtering:

Content Based approach is based on the matching of user profile and some specific characteristics of an item. The principle thought behind CB Recommender Systems is that they proposed things that are like the ones previously evaluated with a high appraising by the objective purchaser.

The tech giants in the various fields of online business, entertainment, shopping like Amazon, Flipkart, Netflix, Alibaba. These organizations utilize distinctive recommendation systems, for example, content-based sifting, synergistic and different calculations. Netflix utilizes Netflix Recommendation System (NRE) which is content-dependent on every individual client profile. The motor channels throughout 3,000 titles all at once utilizing 1,300 proposal groups dependent on client inclinations. It's so exact that 80% of Netflix watcher action is driven by customized suggestions from the motor. It's assessed that the NRE saves Netflix more than $1 billion every year.

E-commerce organizations like Amazon, Flipkart utilize distinctive recommendation systems to give a newly added recommendation to clients. Amazon utilizes at present thing shared separating, which scales to colossal datasets and produces an incredible suggest dynamically. This technique is a sort of data separating framework which predicts the "rating" or inclinations that the customer is keen on.

The beginnings of the substance-based proposal technique can be followed back to data recovery and separating research. Thusly, it very well may be considered as a characteristic expansion and continuation of data separating. However, the CB recommendations outperform typical information retrieval methods because, when information is acquired for the user, user preferences are also taken into consideration.

A Content-Based system categorizes items of interest based on their salient characteristics. A CB recommender creates a user profile based on the characteristics of the things the user has previously evaluated. To create user profiles, a variety of learning approaches such as decision trees, neural networks, and vector-based representations can be used.

The progression of a Content-Based Recommendation System can be summed up with underneath figure:

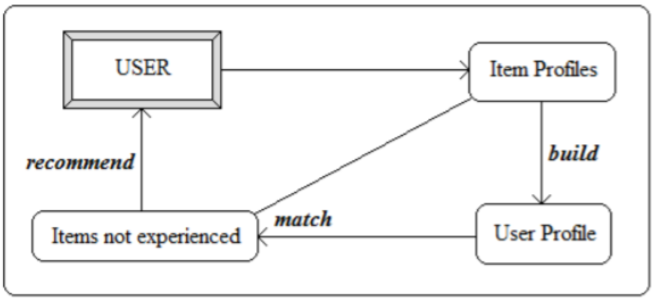


Fig: Workflow of a Content-based Recommendation System.

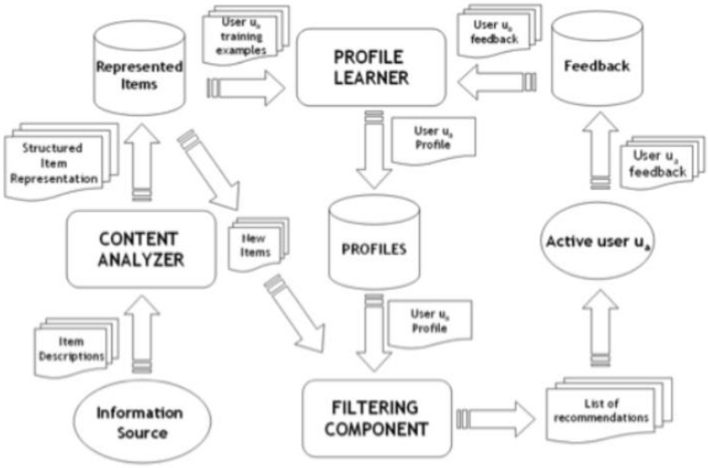
CB systems have the flaw of being unable to account for community endorsements. For example, it might suggest "The Mexican" to a user who enjoys Brad Pitt and Julia Roberts, even though many of their peers despise the picture. However, it has an advantage over the CF technique in that CB systems can recommend new things to users who have never utilize the system before, whereas collaborative systems cannot.

Architecture of a Content-Based Recommender:

Content-Based Recommender Systems are determined on the standard of suggesting things dependent on their substance.

The three principal components are:

- A Content Analyzer, that provides us with a characterization of the things, utilizing a type of portrayal.
- A Profile Learner makes a profile that addresses every client's inclination.
- A Filtering Component, that takes every one of the info esteems and makes the rundown of proposals for every client.



•

Figure: Architecture of a Content-based Recommender.

METHODOLOGY:

Many of the strategies are used in the information filtering approach, which is based on the information retrieval domain. As far as the clients' advantages, data separating varies from data recovery in one manner. While clients present inquiries in data recovery, clients in data separating have profiles that portray their drawn-out interests, and the filtering system strives to give appropriate items to each user on a long run basis. As previously stated, user details and item details can both be made up of a set of phrases. The filtering algorithm chooses and ranks-orders the relevant items and presents them to the user based on some measure of common attributes between the individual profiles.

User input, whether explicit or implicit, can determine the real relevance of an item delivered by the system to a user. Express criticism requires the client to pronounce the level of importance of the conveyed thing, while implied input derives pertinence from the client's conduct, like understanding time. The user may prefer implicit feedback, but it is more complex to create and less accurate. User feedback allows you to change your profile based on what you've read, liked, and disliked.

In terms of information filtering, there are two basic approaches: collaborative and content based. In collaborative filtering, the system chooses and ranks items for a user depending on how similar the user is to other clients who have previously read/liked similar stuff. The framework picks and positions the things dependent on the comparability of the client's profile and the profiles of the qualities in content-based separating.

We will present content-based filtering using cosine similarity in this project, which we have implemented to recommend movies.

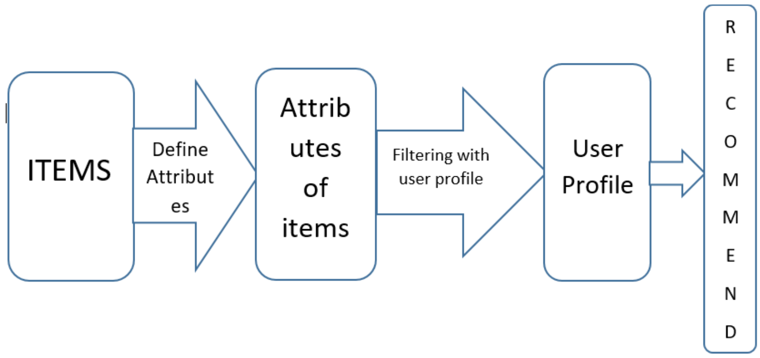


Figure: Flow Diagram of Content-Based Filtering.

IMPLEMENTATION

In this notebook, we will explore Content-based recommendation systems and implement a simple version of one using Python and the Pandas library.

Let's set maximum rows to be displayed at any time to not more than 20.

Reading the data to the data frame.

```
In [1]: import pandas as pd

In [2]: pd.set_option('max_rows', 20)

In [5]: movies_data=r'C:\Users\chara\Downloads\MovieRecommend\movies.csv'

In [6]: ratings_data=r'C:\Users\chara\Downloads\MovieRecommend\ratings.csv'

In [8]: missing_values = ['na','--','?','-','None','none','non']

In [9]: movies_df = pd.read_csv(movies_data, na_values=missing_values)
        ratings_df = pd.read_csv(ratings_data, na_values=missing_values)

In [10]: print('Movies_df Shape:',movies_df.shape)
         movies_df

Movies_df Shape: (9742, 3)
```

Out[10]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
9737	193581	Black Butler: Book of the Atlantic (2017)	Action|Animation|Comedy|Fantasy
9738	193583	No Game No Life: Zero (2017)	Animation|Comedy|Fantasy
9739	193585	Flint (2017)	Drama
9740	193587	Bungo Stray Dogs: Dead Apple (2018)	Action|Animation
9741	193609	Andrew Dice Clay: Dice Rules (1991)	Comedy

9742 rows × 3 columns

Eliminating the year from the title segment and spot it in its own segment, utilizing the convenient concentrate capacity of pandas, close by python regex.

We indicate the enclosures, so we don't struggle with films that have a long time in their titles.

Utilizing customary articulations to observe a year put away between enclosures

Removing the years from the 'title' column

```
In [11]: print('Ratings_df Shape:',ratings_df.shape)
         ratings_df.head()
```

```
Ratings_df Shape: (100836, 4)
```

Out[11]:

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

```
In [12]: movies_df['year'] = movies_df.title.str.extract('(\(\d\d\d\d\))',expand=False)
         movies_df.head(3)
```

Out[12]:

	movieId	title	genres	year
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	(1995)
1	2	Jumanji (1995)	Adventure|Children|Fantasy	(1995)
2	3	Grumpier Old Men (1995)	Comedy|Romance	(1995)

```
In [13]: movies_df['year'] = movies_df.year.str.extract('(\d\d\d\d)',expand=False)
         movies_df.head(3)
```

Out[13]:

	movieId	title	genres	year
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1995
1	2	Jumanji (1995)	Adventure|Children|Fantasy	1995
2	3	Grumpier Old Men (1995)	Comedy|Romance	1995

Applying the strip capacity to dispose of any completion whitespace characters that might have shown up.

Split the qualities in the Genres section into a rundown of Genres to improve on future use. This can be fully achieved by applying the Python's split string function on the correct column.

```
In [15]: movies_df['title'] = movies_df['title'].apply(lambda x: x.strip())
         movies_df.head()
```

Out[15]:

	movieId	title	genres	year
0	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	1995
1	2	Jumanji	Adventure|Children|Fantasy	1995
2	3	Grumpier Old Men	Comedy|Romance	1995
3	4	Waiting to Exhale	Comedy|Drama|Romance	1995
4	5	Father of the Bride Part II	Comedy	1995

```
In [16]: movies_df['genres'] = movies_df.genres.str.split('|')
         movies_df.head()
```

Out[16]:

	movieId	title	genres	year
0	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995
1	2	Jumanji	[Adventure, Children, Fantasy]	1995
2	3	Grumpier Old Men	[Comedy, Romance]	1995
3	4	Waiting to Exhale	[Comedy, Drama, Romance]	1995
4	5	Father of the Bride Part II	[Comedy]	1995

```
In [17]: movies_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9742 entries, 0 to 9741
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   movieId  9742 non-null   int64
 1   title    9742 non-null   object
 2   genres   9742 non-null   object
 3   year     9729 non-null   object
dtypes: int64(1), object(3)
memory usage: 304.6+ KB
```

In [25]:
```
movies_df.head(3)
```

Out[25]:

	movieId	title	genres	year
0	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995
1	2	Jumanji	[Adventure, Children, Fantasy]	1995
2	3	Grumpier Old Men	[Comedy, Romance]	1995

In [26]:
```
movies_with_genres = movies_df.copy(deep=True)
x = []
for index, row in movies_df.iterrows():
    x.append(index)
    for genre in row['genres']:
        movies_with_genres.at[index, genre] = 1
print(len(x) == len(movies_df))
movies_with_genres.head(3)
```

True

Out[26]:

	movieId	title	genres	year	Adventure	Animation	Children	Comedy	Fantasy	Romance	...	Horror	Mystery	Sci-Fi	War	Musical	Documentary	I
0	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995	1.0	1.0	1.0	1.0	1.0	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
1	2	Jumanji	[Adventure, Children, Fantasy]	1995	1.0	NaN	1.0	NaN	1.0	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
2	3	Grumpier Old Men	[Comedy, Romance]	1995	NaN	NaN	NaN	1.0	NaN	1.0	...	NaN	NaN	NaN	NaN	NaN	NaN	

3 rows × 24 columns

In [27]:
```
movies_with_genres = movies_with_genres.fillna(0)
movies_with_genres.head(3)
```

Out[27]:

	movieId	title	genres	year	Adventure	Animation	Children	Comedy	Fantasy	Romance	...	Horror	Mystery	Sci-Fi	War	Musical	Documentary	I
0	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995	1.0	1.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
1	2	Jumanji	[Adventure, Children, Fantasy]	1995	1.0	0.0	1.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
2	3	Grumpier Old Men	[Comedy, Romance]	1995	0.0	0.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	

3 rows × 24 columns

In [28]:
```
print('Ratings_df shape:',ratings_df.shape)
ratings_df.head()
```

Ratings_df shape: (100836, 4)

Out[28]:

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

To characterize our proposal work these are the accompanying advances we'll follow:

- Get the file of the film given its title.
- Get the rundown of cosine closeness scores for that specific film with all motion pictures. Convert it into a rundown of tuples where the main component is its position and the second is the comparability score.
- Sort the rundown of tuples dependent on the closeness scores; that is, the second element.
- Get the top 10 elements of this list. Ignore the first element as it refers to self (the movie most similar to a particular movie is the movie itself).
- Return the titles relating to the lists of the top components.

In [15]:

```
# Function that takes in movie title as input and outputs most similar movies
def get_recommendations(title, cosine_sim=cosine_sim):
    # Get the index of the movie that matches the title
    idx = indices[title]

    # Get the pairwsie similarity scores of all movies with that movie
    sim_scores = list(enumerate(cosine_sim[idx]))

    # Sort the movies based on the similarity scores
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

    # Get the scores of the 10 most similar movies
    sim_scores = sim_scores[1:11]

    # Get the movie indices
    movie_indices = [i[0] for i in sim_scores]

    # Return the top 10 most similar movies
    return df2['title'].iloc[movie_indices]
```

5. EXPERIMENTAL RESULTS

CONTENT BASED RECOMMENDER SYSTEM

In [33]:
```
Lawrence_movie_ratings = [
            {'title':'Predator', 'rating':4.9},
            {'title':'Final Destination', 'rating':4.9},
            {'title':'Mission Impossible', 'rating':4},
            {'title':"Beverly Hills Cop", 'rating':3},
            {'title':'Exorcist, The', 'rating':4.8},
            {'title':'Waiting to Exhale', 'rating':3.9},
            {'title':'Avengers, The', 'rating':4.5},
            {'title':'Omen, The', 'rating':5.0}
         ]
Lawrence_movie_ratings = pd.DataFrame(Lawrence_movie_ratings)
Lawrence_movie_ratings
```

Out[33]:

	title	rating
0	Predator	4.9
1	Final Destination	4.9
2	Mission Impossible	4.0
3	Beverly Hills Cop	3.0
4	Exorcist, The	4.8
5	Waiting to Exhale	3.9
6	Avengers, The	4.5
7	Omen, The	5.0

In [34]:
```
Lawrence_movie_Id = movies_df[movies_df['title'].isin(Lawrence_movie_ratings['title'])]
Lawrence_movie_ratings = pd.merge(Lawrence_movie_Id, Lawrence_movie_ratings)
Lawrence_movie_ratings
```

Out[34]:

	movieId	title	genres	year	rating
0	4	Waiting to Exhale	[Comedy, Drama, Romance]	1995	3.9
1	1350	Omen, The	[Horror, Mystery, Thriller]	1976	5.0
2	45662	Omen, The	[Horror, Thriller]	2006	5.0
3	1997	Exorcist, The	[Horror, Mystery]	1973	4.8
4	2153	Avengers, The	[Action, Adventure]	1998	4.5
5	89745	Avengers, The	[Action, Adventure, Sci-Fi, IMAX]	2012	4.5
6	3409	Final Destination	[Drama, Thriller]	2000	4.9
7	3527	Predator	[Action, Sci-Fi, Thriller]	1987	4.9
8	4085	Beverly Hills Cop	[Action, Comedy, Crime, Drama]	1984	3.0

In [35]:
```
Lawrence_movie_ratings = Lawrence_movie_ratings.drop(['genres','year'], 1)
Lawrence_movie_ratings
```

Out[35]:

	movieId	title	rating
0	4	Waiting to Exhale	3.9
1	1350	Omen, The	5.0
2	45662	Omen, The	5.0
3	1997	Exorcist, The	4.8
4	2153	Avengers, The	4.5
5	89745	Avengers, The	4.5
6	3409	Final Destination	4.9
7	3527	Predator	4.9
8	4085	Beverly Hills Cop	3.0

In [36]:
```python
Lawrence_genres_df = movies_with_genres[movies_with_genres.movieId.isin(Lawrence_movie_ratings.movieId)]
Lawrence_genres_df
```

Out[36]:

genres	year	Adventure	Animation	Children	Comedy	Fantasy	Romance	...	Horror	Mystery	Sci-Fi	War	Musical	Documentary	IMAX	Western	Film-Noir	ge li
[Comedy, Drama, Romance]	1995	0.0	0.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
[Horror, Mystery, Thriller]	1976	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
[Horror, Mystery]	1973	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
[Action, Adventure]	1998	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
[Drama, Thriller]	2000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
[Action, Sci-Fi, Thriller]	1987	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
[Action, Comedy, Crime, Drama]	1984	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
[Horror, Thriller]	2006	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
[Action, Adventure, Sci-Fi, IMAX]	2012	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	

In [38]:
```python
print('Shape of Lawrence_movie_ratings is:',Lawrence_movie_ratings.shape)
print('Shape of Lawrence_genres_df is:',Lawrence_genres_df.shape)
```

```
Shape of Lawrence_movie_ratings is: (9, 3)
Shape of Lawrence_genres_df is: (9, 20)
```

In [39]:
```python
Lawrence_profile = Lawrence_genres_df.T.dot(Lawrence_movie_ratings.rating)
Lawrence_profile
```

Out[39]:
```
Adventure             7.8
Animation             0.0
Children              0.0
Comedy                8.8
Fantasy               0.0
Romance               3.9
Drama                13.3
Action               17.2
Crime                 4.9
Thriller             18.9
Horror               14.9
Mystery              10.0
Sci-Fi                7.5
War                   0.0
Musical               0.0
Documentary           0.0
IMAX                  3.0
Western               0.0
Film-Noir             0.0
(no genres listed)    0.0
dtype: float64
```

```
In [40]: movies_with_genres = movies_with_genres.set_index(movies_with_genres.movieId)
         movies_with_genres.head()
```

Out[40]:

	movieId	title	genres	year	Adventure	Animation	Children	Comedy	Fantasy	Romance	...	Horror	Mystery	Sci-Fi	War	Musical	Document
movieId																	
1	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995	1.0	1.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
2	2	Jumanji	[Adventure, Children, Fantasy]	1995	1.0	0.0	1.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
3	3	Grumpier Old Men	[Comedy, Romance]	1995	0.0	0.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	
4	4	Waiting to Exhale	[Comedy, Drama, Romance]	1995	0.0	0.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	
5	5	Father of the Bride Part II	[Comedy]	1995	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	

5 rows × 24 columns

```
In [41]: movies_with_genres.drop(['movieId','title','genres','year'], axis=1, inplace=True)
         movies_with_genres.head()
```

Out[41]:

	Adventure	Animation	Children	Comedy	Fantasy	Romance	Drama	Action	Crime	Thriller	Horror	Mystery	Sci-Fi	War	Musical	Documentary	IMA:
movieId																	
1	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
2	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
3	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
4	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
5	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.

```
In [43]: recommendation_table_df.sort_values(ascending=False, inplace=True)
         recommendation_table_df.head(20)
```

```
Out[43]: movieId
81132    0.869328
43932    0.742287
7235     0.707804
36509    0.692377
79132    0.678766
51545    0.663339
26701    0.651543
198      0.651543
6395     0.651543
54771    0.651543
74685    0.651543
4956     0.634301
60684    0.634301
4210     0.627949
71999    0.622505
31804    0.621597
2617     0.613430
72165    0.613430
91542    0.613430
27683    0.610708
dtype: float64
```

In [44]:
```
copy = movies_df.copy(deep=True)
copy = copy.set_index('movieId', drop=True)
top_20_index = recommendation_table_df.index[:20].tolist()
recommended_movies = copy.loc[top_20_index, :]
recommended_movies
```

Out[44]:

movieId	title	genres	year
81132	Rubber	[Action, Adventure, Comedy, Crime, Drama, Film...	2010
43932	Pulse	[Action, Drama, Fantasy, Horror, Mystery, Sci-...	2006
7235	Ichi the Killer (Koroshiya 1)	[Action, Comedy, Crime, Drama, Horror, Thriller]	2001
36509	Cave, The	[Action, Adventure, Horror, Mystery, Sci-Fi, T...	2005
79132	Inception	[Action, Crime, Drama, Mystery, Sci-Fi, Thrill...	2010
51545	Pusher III: I'm the Angel of Death	[Action, Comedy, Drama, Horror, Thriller]	2005
26701	Patlabor: The Movie (Kidô keisatsu patorebâ: T...	[Action, Animation, Crime, Drama, Film-Noir, M...	1989
198	Strange Days	[Action, Crime, Drama, Mystery, Sci-Fi, Thriller]	1995
6395	Crazies, The (a.k.a. Code Name: Trixie)	[Action, Drama, Horror, Sci-Fi, Thriller]	1973
54771	Invasion, The	[Action, Drama, Horror, Sci-Fi, Thriller]	2007
74685	Crazies, The	[Action, Drama, Horror, Sci-Fi, Thriller]	2010
4956	Stunt Man, The	[Action, Adventure, Comedy, Drama, Romance, Th...	1980
60684	Watchmen	[Action, Drama, Mystery, Sci-Fi, Thriller, IMAX]	2009
4210	Manhunter	[Action, Crime, Drama, Horror, Thriller]	1986
71999	Aelita: The Queen of Mars (Aelita)	[Action, Adventure, Drama, Fantasy, Romance, S...	1924
31804	Night Watch (Nochnoy dozor)	[Action, Fantasy, Horror, Mystery, Sci-Fi, Thr...	2004
2617	Mummy, The	[Action, Adventure, Comedy, Fantasy, Horror, T...	1999
72165	Cirque du Freak: The Vampire's Assistant	[Action, Adventure, Comedy, Fantasy, Horror, T...	2009
91542	Sherlock Holmes: A Game of Shadows	[Action, Adventure, Comedy, Crime, Mystery, Th...	2011
27683	Tremors 4: The Legend Begins	[Action, Comedy, Horror, Sci-Fi, Thriller, Wes...	2004

Suggestions of comparable films from the past film information.

In [16]:
```
get_recommendations('The Dark Knight Rises')
```

Out[16]:
```
65                              The Dark Knight
299                              Batman Forever
428                              Batman Returns
1359                                     Batman
3854    Batman: The Dark Knight Returns, Part 2
119                               Batman Begins
2507                                  Slow Burn
9            Batman v Superman: Dawn of Justice
1181                                        JFK
210                              Batman & Robin
Name: title, dtype: object
```

In [17]:
```
get_recommendations('The Avengers')
```

Out[17]:
```
7               Avengers: Age of Ultron
3144                            Plastic
1715                            Timecop
4124                 This Thing of Ours
3311              Thank You for Smoking
3033                      The Corruptor
588     Wall Street: Money Never Sleeps
2136         Team America: World Police
1468                       The Fountain
1286                        Snowpiercer
Name: title, dtype: object
```

In [30]:
```
get_recommendations('The Godfather', cosine_sim2)
```

Out[30]:
```
867      The Godfather: Part III
2731      The Godfather: Part II
4638    Amidst the Devil's Wings
2649           The Son of No One
1525              Apocalypse Now
1018             The Cotton Club
1170     The Talented Mr. Ripley
1209              The Rainmaker
1394               Donnie Brasco
1850                   Scarface
Name: title, dtype: object
```

6. DISCUSSION OF RESULTS

A section of the Each Movie dataset was used. This dataset contains 610 people who were chosen at random and 100836 movie ratings for which IMDb content was accessible. We have columns like movie_id, title, genre, and year in the Movies data set, and user_id, movie_id, and rating in the Ratings data set. We isolated the rating dataset into tests and prepared sets to evaluate different separating calculations. A subset of the ratings data from the Each movie dataset is used in the rating database. Utilizing a for the most part utilized mistake measure, the preparation set is used to anticipate evaluations in the test set.

The proposed method is used by a single user. Here, we use the weighted average of every movie based on his profile as well as the offer of the top twenty films that correspond to his tastes.

Out[0]:

movieId	title	genres	year
81132	Rubber	[Action, Adventure, Comedy, Crime, Drama, Film...	2010
43932	Pulse	[Action, Drama, Fantasy, Horror, Mystery, Sci-...	2006
7235	Ichi the Killer (Koroshiya 1)	[Action, Comedy, Crime, Drama, Horror, Thriller]	2001
36509	Cave, The	[Action, Adventure, Horror, Mystery, Sci-Fi, T...	2005
79132	Inception	[Action, Crime, Drama, Mystery, Sci-Fi, Thrill...	2010
51545	Pusher III: I'm the Angel of Death	[Action, Comedy, Drama, Horror, Thriller]	2005
198	Strange Days	[Action, Crime, Drama, Mystery, Sci-Fi, Thriller]	1995
6395	Crazies, The (a.k.a. Code Name: Trixie)	[Action, Drama, Horror, Sci-Fi, Thriller]	1973
54771	Invasion, The	[Action, Drama, Horror, Sci-Fi, Thriller]	2007
74685	Crazies, The	[Action, Drama, Horror, Sci-Fi, Thriller]	2010
26701	Patlabor: The Movie (Kidô keisatsu patorebâ: T...	[Action, Animation, Crime, Drama, Film-Noir, M...	1989
4956	Stunt Man, The	[Action, Adventure, Comedy, Drama, Romance, Th...	1980
60684	Watchmen	[Action, Drama, Mystery, Sci-Fi, Thriller, IMAX]	2009
4210	Manhunter	[Action, Crime, Drama, Horror, Thriller]	1986
71999	Aelita: The Queen of Mars (Aelita)	[Action, Adventure, Drama, Fantasy, Romance, S...	1924
31804	Night Watch (Nochnoy dozor)	[Action, Fantasy, Horror, Mystery, Sci-Fi, Thr...	2004
2617	Mummy, The	[Action, Adventure, Comedy, Fantasy, Horror, T...	1999
72165	Cirque du Freak: The Vampire's Assistant	[Action, Adventure, Comedy, Fantasy, Horror, T...	2009
91542	Sherlock Holmes: A Game of Shadows	[Action, Adventure, Comedy, Crime, Mystery, Th...	2011
27683	Tremors 4: The Legend Begins	[Action, Comedy, Horror, Sci-Fi, Thriller, Wes...	2004

Figure: Recommended Movies by the user

These are the top 20 movies that the client has recommended, and we can observe the movie_id, title, genre, and year of release in the output.

When compared to CBF and CF, we can observe that there are three sorts of filtering methods in which hybrid filtering accuracy is good, i.e., at the point when content-based and communitarian sifting strategies are joined.

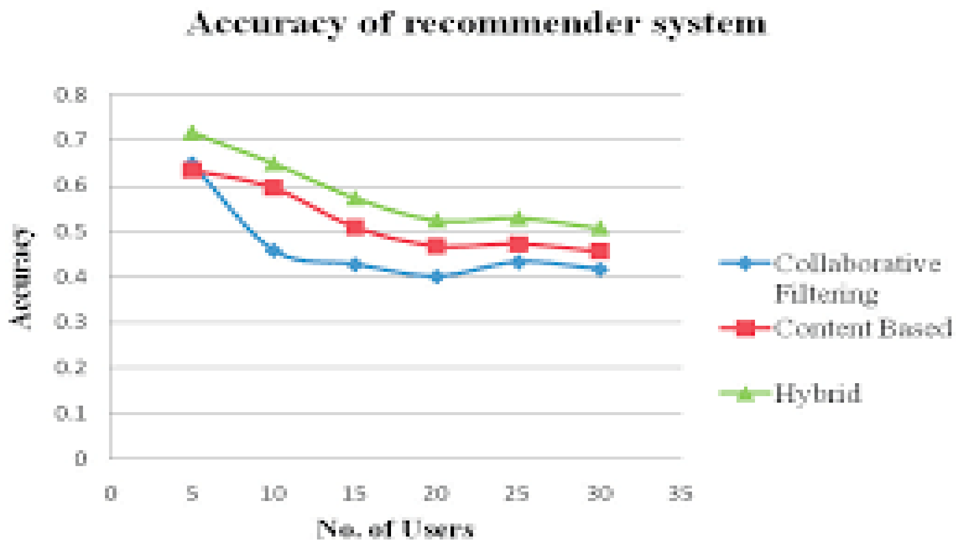


Figure: Accuracy of recommender system

We can also observe in the graph that:

1. Because Collaborative Filtering relies on overlap in ratings across users, the accuracy of the technique has deteriorated due to sparsity in ratings, i.e., a small group of people evaluating the same objects.

2. Because of its capacity to forecast out-of-box recommendations, Collaborative Filtering sometimes produced more relevant results.

3. Segment factors used to channel Content-based ideas have worked on the procedure's exactness.

FUTURE SCOPE:

Content-Based Filtering recommends any product or service based on similarities between products and the user's previous activity on the website. Because it doesn't rely on other users' input, this filter can suggest products based on a similarity factor, which helps to avoid a cold start for any new products. However, content-based filtering necessitates an ample of subject expertise in order for the recommendations to be completely accurate. Collaborative filtering, On the other side, collaborative filtering tries to address the shortcomings of content-based filtering Rather than focusing on a single user, the community sifting framework considers all clients and gatherings them as indicated by their inclinations. Basically, it recommends an item 'x's to client 'a' dependent on the inclinations of client 'b'; users 'a' and 'b' must have shared interests in the past, which is why they are grouped together. Collaborative filtering requires less topic expertise, provides more accurate recommendations, and can adjust to changing user preferences over time. Building a cooperative separating framework is in like manner very expensive and furthermore tedious.

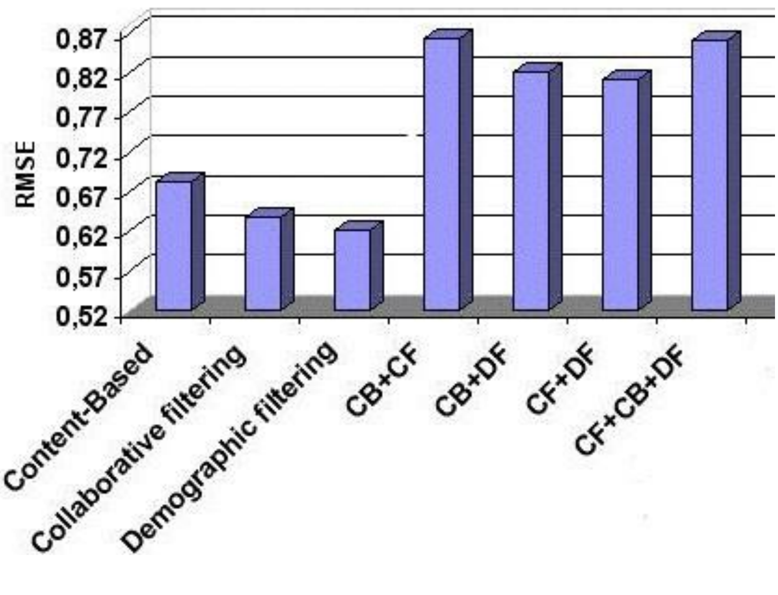


Figure: Comparison of recommender systems

Content-based filtering is more accurate than collaborative and demographic filtering, as represented in the graph above, but only when there are fewer users and a large amount of data.

Nonetheless, we can blend content-based and cooperative-based separating to make a crossbreed sifting that is considerably more exact than content-based sifting. Combining CF and CBF, as well as DF+CF, DF+CBF, and three filtering algorithms, can result in a very accurate recommender system. So, rather than using single filtering methods, we can combine and employ a combination of filtering methods to increase efficiency and accuracy.

7. SUMMARY & CONCLUSIONS

E-commerce organizations like Amazon, Flipkart utilize distinctive recommendation systems to give a newly added recommendation to clients Amazon utilizes presently thing synergistic sifting, which scales to immense datasets and produces an incredible suggest dynamically. This strategy is a sort of data sifting framework which foresees the "rating" or inclinations that the customer is keen in.

The starting points of the substance-based suggestion system can be followed back to data recovery and sifting research. Consequently, it can be thought of as a natural extension and continuation of information filtering. However, the CB recommendations outperform typical information retrieval methods because, when information is acquired for the user, user preferences are also taken into consideration.

Recommender frameworks are ending up a significant instrument for giving ideas to clients as required on their necessities. They can provide strategic advantages to businesses who use them, as well as increased consumer pleasure and loyalty. If a company does not apply RS, it is more likely to be forced out of the market. We explore three different forms of RS in this work, but it's impossible to say that one is better than the remaining because simple systems can be cheaper to develop but provide poorer accuracy. However, almost every RS will have issues when it is first introduced because there is insufficient data on users and objects. When first adopting the RS, it is critical to enhance accuracy because poor user recommendations might lessen the impact of the RS on sales.

To achieve this accuracy, most memory-based methods and calculations were planned and refined here and there (e.g., kNN measurements, particular worth deterioration, and so forth) At the present time, half and half methodologies (principally collective sifting and content separating) are being utilized to work on the nature of the proposals. To accomplish this accuracy, most memory-based methods and calculations were planned and refined somehow or another (e.g., kNN measurements, solitary worth disintegration, and so on) At the present time, crossbreed draws near (primarily cooperative separating and content sifting) are being utilized to work on the nature of the proposals. In the subsequent advance, calculations that acknowledged social contribution with before cross breed procedures were obliged and created (e.g., trust-mindful calculations, social versatile methodologies, interpersonal organizations investigation, and so on). Area information is by and by being incorporated into existing proposal frameworks utilizing half-breed calculations. Future exploration will focus on overhauling existing methodologies and calculations to work on the nature of recommender frameworks. For example, novel research lines will be developed in the following fields:

(1) Existing proposal frameworks that utilize different sorts of accessible information will be consolidated in a legitimate way.

(2) Processes for recommender systems ensure security and privacy. (3) Machine-controlled analysis of diverse data is facilitated by flexible frameworks.

8. REFERENCES

[1]. Recommender Systems Prem Melville and Vikas Sindhwani IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 {pmelvil, vsindhw}@us.ibm.com

[2]. Recommender Systems: Types of Filtering Techniques Iateilang Ryngksai, Chameikho Dept. of Computer Science & Engineering and IT Don Bosco College of Engineering. International Journal of Engineering Research & Technology (IJERT) IJERT ISSN: 2278-0181 Vol. 3 Issue 11, November-2014.

[3]. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-9S, July 2019 373 Retrieval Number: I10590789S19/19©BEIESP DOI: 10.35940/ijitee.I1059.0789S19 Published By: Blue Eyes Intelligence Engineering & Sciences Publication Techniques of Recommender System Harleen kaur, Gourav Bathla.

[4]. Using Content-Based Filtering for Recommendation1 1 This research has been supported by NetlinQ Robin van Meteren1 and Maarten van Someren2 1 NetlinQ Group, Gerard Brandtstraat 26-28, 1054 JK, Amsterdam, The Netherlands, robin@netlinq.nl 2 University of Amsterdam, Roeterstraat 18, The Netherlands, marten@swi.psy.uva.nl.

[5]. Content-based filtering for recommendation systems using multiattribute networks. JieunSon, Seoung BumKim, Department of Industrial Management Engineering, Korea University, 145 Anam-Ro, Seoungbuk-Gu, Anam-dong, Seoul 136-713, South Korea.

[6]. AN ONTOLOGY- CONTENT-BASED FILTERING METHOD Peretz Shoval, Veronica Maidel, Bracha Shapira, International Journal "Information Theories & Applications" Vol.15 / 2008.

[7]. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. Poonam B. Thorat, R. M. Goudar, Sunita Barve Computer Engineering MIT

Academy of Engineering Pune India, 4, January 2015.

[8]. Content-Based Collaborative Filtering for News Topic Recommendation Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie and Qiang Yang Hong Kong University of Science and Technology, Hong Kong Renmin University of China, Beijing, China Microsoft Research, Beijing, China.

[9]. Content-based book recommending using learning for text categorization. Raymond J. Mooney, Loriene Roy.

[10]. Comparing Content Based and Collaborative Filtering in Recommender Systems Parul Aggarwal, Vishal Tomar, Aditya Kathuria. International Journal of New Technology and Research (IJNTR) ISSN:2454-4116, Volume-3, Issue-4, April 2017.

- [11]. Movie Recommendation System Using Genome Tags and Content-Based Filtering. Syed M. Ali, Gopal K. Nayak, Rakesh K. Lenka, Rabindra K. Barik.

[12]. A Method for Weighting Multi-valued Features in Content-Based Filtering. Manuel J. Barranco, Luis Martínez

[13]. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. Chumki Basu, cbasu@bellcore.com, Haym Hirsh-hirsh@cs.rutgers.edu, William Cohen - wcohen@research.att.com

[14]. Hybrid Collaborative Filtering and Content-Based Filtering for Improved Recommender System. Kyung-Yong Jun, Dong-Hyun Park, Jung-Hyun Lee.

[15]. Hybrid Collaborative Filtering and Content-Based Filtering for Improved Recommender System. Kyung-Yong Jung, Dong-Hyun Park and Jung-Hyun Lee, Department of Computer Science and Engineering. Inha University, Korea.

[16]. Joining Collaborative and Content-based Filtering Patrick Baudisch Integrated Publication and Information Systems Institute IPSI German National Research Center for

Information Technology GMD 64293 Darmstadt, Germany +49-6151-869-854 baudisch@gmd.de

[17]. Content-based Recommender Systems: State of the Art and Trends Pasquale Lops, Marco de Gemmis and Giovanni Semeraro.

[18]. A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System. Geetha G, Safa M, Fancy C, Saranya D, Department of Information Technology, SRM Institute of Science and Technology 4 Department of Electronics and Communication Engineering, Arasu Engineering College.

[19]. A CONTENT BASED MOVIE RECOMMENDATION SYSTEM EMPOWERED BY COLLABORATIVE MISSING DATA PREDICTION BY HİLAL KARAMAN.

[20]. Manoj Kumar, D.K Yadav, Ankur Singh, Vijay Kr. Gupta," A Movie Recommender System: MOVREC" International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3, August 2015.

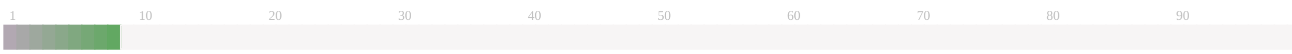
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

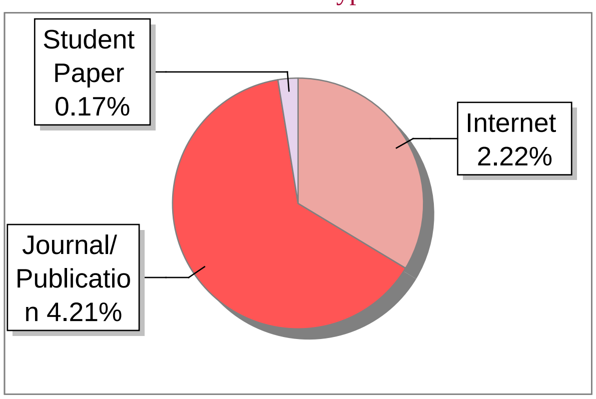
Author Name	KOUSHIK
Title	COSINE
Paper/Submission ID	415314
Submission Date	2021-11-23 16:36:56
Total Pages	46
Document type	Article

Result Information

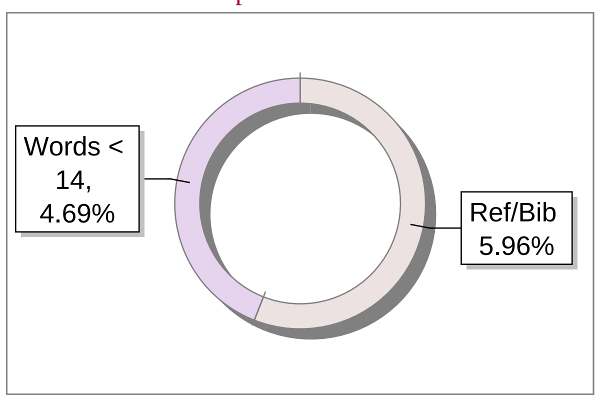
Similarity **9 %**



Sources Type



Report Content



Alert..! Improper usage of 'Quotes' work in a report

Exclude Information

References/Bibliography	Not Excluded
Quotes	Not Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	**0 %**

A Unique QR Code use to View/Download/Share Pdf File

DrillBit
Anti Plagiarism Software

DrillBit Similarity Report

9 SIMILARITY %	**45** MATCHED SOURCES	**A** GRADE	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	towardsdatascience.com	2	Internet Data
2	iopscience.iop.org	1	Publication
3	www.ijeat.org	1	Publication
4	From margins to mainstream Private school outreach inclusion processe by Laur-2005	<1	Publication
5	Effective personalized recommendation based on time-framed navigation clustering by Feng-Hs-2004	<1	Publication
6	research.aota.org	<1	Internet Data
7	II M Chevaliers Tableau de la Plaine de Troye illustrated and confirmed, from by Dalzel-1798	<1	Publication
8	Automatically solving two-variable linear algebraic word problems using text min by Rehman-2018	<1	Publication
9	IdentityBased Encryption with Cloud Revocation Authority and Its Applications -Submited to JNTUH,TELANGANA By '15P61D5820'	<1	Student Paper
10	An Overview of Universal Information Services Concepts and Technologi by Heiber-1989	<1	Publication
11	A family of metrics for biopolymers based on counting independent sets by J-2005	<1	Publication

12	The Fortune of Machiavellis Unarmed Prophet by Scott-2018	<1	Publication
13	www.dx.doi.org	<1	Publication
14	Time and the River A History of Cleburne Countyby Evalena Berry by Revie-1983	<1	Publication
15	Thesis submitted to dspace.mit.edu	<1	Publication
16	Talking to cancer patients about complementary therapies Is it the physicians by Verhoef-2008	<1	Publication
17	Leukocyte-Dependent Regulation of Cardiac Fibrosis by Okyere-2020	<1	Publication
18	Guidelines for perioperative care after radical cystectomy for bladder by Cerantola-213	<1	Publication
19	Consciously Thinking About Consciousness by Myro-2004	<1	Publication
20	www.whatech.com	<1	Internet Data
21	The Doctrines of Public Policy and Competence in InvestorState Arbit, by Ayad, Mary B.- 2013	<1	Publication
22	epdf.pub	<1	Internet Data
23	Literary and Rhetorical Theory in Irenaeus, Part 2, by Briggman, Anthony- 2016	<1	Publication
24	Exploiting reviewers comment histories for sentiment analysis by Basiri-2014	<1	Publication
25	www.india-astrologer.com	<1	Internet Data
26	www.ijsdr.org	<1	Publication
27	www.holyorderofmans.org	<1	Internet Data
28	www.gyanvihar.org	<1	Publication

29	www.dx.doi.org	<1	Publication
30	www.dx.doi.org	<1	Publication
31	Thesis submitted to shodhganga - shodhganga.inflibnet.ac.in	<1	Publication
32	The impact of land use change and check-dams on catchment sediment yield by Carolin-2008	<1	Publication
33	SB MAN POWER PLANNING WHIRLPOOL LTD BY 16RQ1E0001, YR 2020, JNTUH	<1	Student Paper
34	Preparation, characterisation and structure of Ti and Al ultrathin oxide films o by Wu-2009	<1	Publication
35	New logarithmic operational laws and their aggregation operators for Pythagorean by Garg-2018	<1	Publication
36	Matter-Induced Neutrino Oscillation in Double Universal Seesaw Model by Sogami-1992	<1	Publication
37	hozir.org	<1	Internet Data
38	Changes in haematological and clinicochemical profiles in blood of apparently he by Elbers-1991	<1	Publication
39	Challenging Authority by Hurd-1991	<1	Publication
40	bmcbioinformatics.biomedcentral.com	<1	Publication
41	IEEE 2018 International Conference on Advances in Big Data, Computi, by Adeyeye, Kehinde N- 2018	<1	Publication
42	IEEE 2013 Humaine Association Conference on Affective Computing and by	<1	Publication
43	www.dx.doi.org	<1	Publication
44	www.arxiv.org	<1	Publication

45 Comet Assay Measurements of DNA Damage in Cells by Laser Microbeams and Trapping by Mohanty-2002 <1 Publication