MOVIE RECOMMENDATION SYSTEM

A

Project Report Submitted for the

Partial fulfillment of B.Tech

Degree in

Computer Science & Engineering

By

Apoorv Bansal(1805213013)

Uttam Singh(1805213062)

Vanshika Gupta(1805213063)

Under the supervision of

Mr. Natthan Singh
Mr. Sandeep Kumar Yadav

Department of Computer Science and Engineering

Institute of Engineering and Technology

Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh.

June, 2022

Contents

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our belief and knowledge, it contains no material previously published or written by another person or material which to a substantial error has been accepted for the award of any degree or diploma of university or other institute of higher learning, except where the acknowledgement has been made in the text. The project has not been submittedby us at any other institute for the requirement of any other degree.

Submitted by: -

Date:

- Name: Apoorv Bansal

  Roll No.: 1805213013

  Branch: IT

  Signature:

- Name: Uttam Singh

  Roll No.: 1805213062

  Branch: IT

  Signature:

- Name: Vanshika Gupta

  Roll No.: 1805213063

  Branch: IT

  Signature:

CERTIFICATE

This is to certify that the project report entitled "**MOVIE RECOMMENDATION SYSTEM**" presented by **, Apoorv Bansal, Uttam Singh and Vanshika Gupta** in the partial fulfillment for the award of Bachelor of Technology in Information technology, is a record of work carried out by them under my supervision and guidance at the Department of Computer Science and Engineering at Institute of Engineering and Technology, Lucknow.

It is also certified that this project has not been submitted at any other Institute for the award of any other degrees to the best of my knowledge.

Mr. Natthan Singh

Mr. Sandeep Kumar Yadav

Department of Computer Science and Engineering

Institute of Engineering and Technology, Lucknow

Acknowledgement

We are very grateful to Mr. Natthan Singh and Mr. Sandeep Kumar Yadav who gave us a chance to work on this project. We would like to thank them for giving us valuable suggestions and ideas.

We would also like to thank our college for providing us all the necessary resources for the project. All in all, we would like to thank everyone involved in this project and helped us with their suggestions to make the project better.

Finally, we would like to thank our parents and friends for always being with us and supporting us in every situation.

Regards,
Apoorv Bansal
Uttam Singh
Vanshika Gupta

Abstract

In today's world because of Covid pandemic everyone even rural population is aware of e-commerce website and e-shopping . There is an exponential growth in both the no.of user and no .of items on e-commerce websites and even on OTT platforms.

Thus finding an desired item in such a huge data set is a very complicated job So make our job easy , Amazon first started recommendation system for there websites and there prime videos OTT platform.

Because of its strength and accuracy in delivering increased suggestions of services and commodities, a movie recommendation system is a vital aspect of our lives. In this research, we offer a movie recommendation system that can recommend movies to both the user and others. Typically, movie recommendation systems estimate what movies a customer would enjoy based on the characteristics of previously enjoyed films.

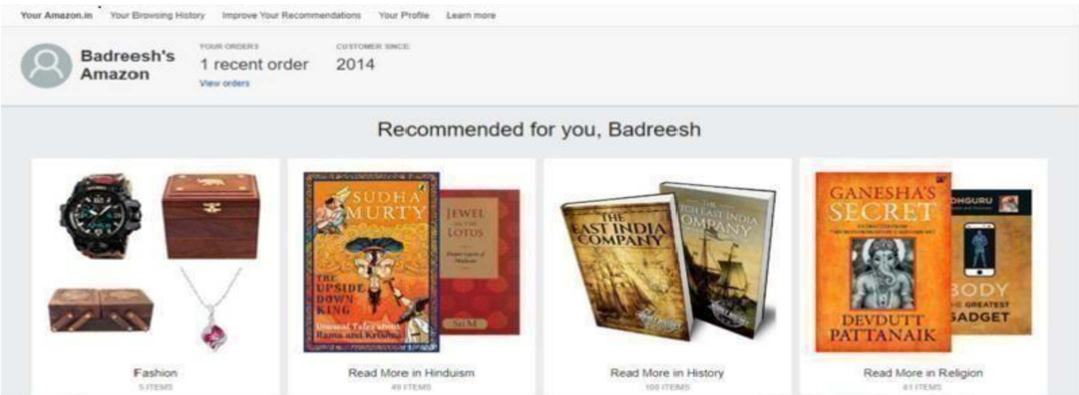
LIST OF FIGURES

Motivation

Motivation is something that drives the people to achieve the desired goals. For Movie recommendation system project , motivation is search of a desired movie with high similarity in dataset of thousand movies. Other than movies, project can be used for different items such as musics , books , electronic items .

AMAZON

When we look for or buy a product or service on Amazon.com, we are shown recommended products based on our previous experiences/searches on the site.



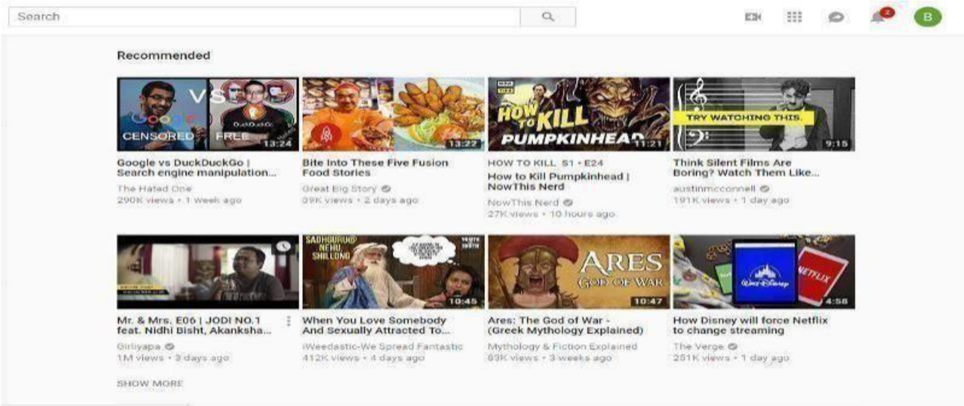
We have already searched for a product or service connected to books and e-books, as illustrated in the diagram above, but Amazon.com categorises it into parts based on previous search experiences.

FACEBOOK & INSTAGRAM

Facebook.com, dubbed Metaverse recently, use recommendation systems on a much larger scale to propose friends, articles, feeds, and news.

YOUTUBE

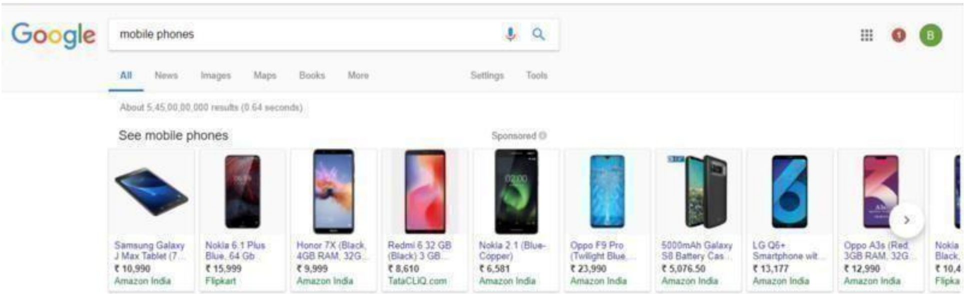
YouTube.com suggests videos based on your prior searches, likes/dislikes, and other variables.



The first video, as shown above, is about Google.com. Despite the fact that I've already seen the video, YouTube.com recommends that I view it again. YouTube.com basically understands what other people like and dislike.

GOOGLE

When we log in to Google.com or search for goods/services, it saves our history and recommends goods/services based on our previous experiences. Advertising accounts for more than 80% of the company's profit, which is unsurprising. Relevant searches also take the customer's geographic location into account.



From the above we can see we were targeted with these ads based on recent search results.

+-

Chapter 1

Introduction

As the WWW grows in size and complexity, the size and complexity of the many datasets, which comprise a variety of factors, grows as well. The Internet currently contains a vast amount of data, the majority of which is uninteresting to the user, either because it is unwelcome (advertisements, spam, etc.) or because it is irrelevant to his interests. However, each user has distinct and distinct interests that may or may not be related to a small percentage of the Web's material. As a result, people are finding it more difficult and time consuming to access information that is relevant to them. The Web can be customized to assist users in obtaining material that is relevant to their past similarity using various recommender systems.

Everyone has recently purchased a commercial product using e-shopping technology that connects businesses and customers (is it also referred as B2C). Consider the purchase of an item from an e-commerce store. A user logs into a website and then buys or searches for things online, rating them according to his or her likes and interests. We can utilize this information, along with the item's representation, to build a user profile and subsequently recommend things that are relevant to his or her interests. A recommendation system is the underlying technology that enables this functionality.

Recommendation systems are a type of information filtering system that primarily aims to assist diverse users in their decision-making process based on their previous experiences and similarity while engaging with a large amount of data. They suggest numerous items of resemblance to consumers depending on preferences they have already expressed, either directly or implicitly. Because of the ever-increasing volume and complexity of data on the internet, such systems have become crucial tools for guiding users in a number of applications, most notably information seeking for e-commerce operations. By exposing users to the most wanted products and providing novelty and insight, recommendation systems assist users overcome various problems of information overload. Technology for recommendation system.

Recommendation system technology is used in a variety of ways by major e- commerce sites like Amazon and Youtube. Many more competitors are on their way, and businesses are vying with one another to figure out the best and most efficient wayto employ this technology, allowing them to provide more accurate service to their customers.

Major sites such as Flipkart, Amazon, Prime or Netflix are using recommendation system technology in various number of different ways based on their availability and customer reviews. Many adopters and competitors are on their way to start a new generation and enterprises are competing with each other in order to find the correct and effective way to use this technology to uplightment of humankind, thus providinga more effective service to their users. The recommendation systems have become an important area now worldwide because customer experience is the main area of concern; the first research paper on the collaborative filtering approach (one of the recommendation system techniques) appeared during the mid-1990s. Both academia and industies have worked together to develop various different approaches and contribute to the recommender systems for improved quality and better experiences from another end during the last decade. A recommendation system is a type of information filtering system, which filters items based on customers interests. From the last few years, the recommendation system became an unavoidable and important part of e- commerce and social websites due to the problem of exponential information overload. In an era of information overloading, recommendation systems has developed for discovering the interesting item according to the individual preference or choice made by them..

Chapter 2

Literature Review

The majority of Movie Recommendation Systems in today's world are built on collaborative filtering methodologies that employ individual prior experiences to provide current results. Collaborative filtering is a strategy that uses information provided by the user based on their surfing history to make decisions. That data is examined and observed, and then a comparable film is recommended to the individuals, which is ordered with the highest-rated film first or whatever scenario the algorithm has chosen..

Luis M Capos investigated, examined, and compared two typical and conventional techniques to recommender systems: content-based filtering and collaborative filtering.
Because they both have flaws, he presented a novel approach that combines Bayesian network and collaborative filtering algorithms.

Harpreet Kaur et al. have proposed a novel model called as the hybrid model. The system is designed to bring together content and collaborative filtering algorithms in one frame,allowing them to overcome each other's constraints. When making recommendations to individuals, the context (such as production, genre, language, and kind...) of the films is also taken into account..

User to User relationships seeks to discover similarities between users based on their profile interactions, such as ratings and reviews. Another sort of similarity is user-item relationships, which look for similarities between a user and an item that the user has purchased in the past and saved in the Web.

Urszula Kuelewska developed a clustering methodology for dealing with recommender systems as one of the methods. For computing cluster representatives, two models have been developed, and both have been assessed. When a Centroid-based solution is combined with memory-based collaborative filtering, it can be used to compare the effectiveness of the two methods.

Costin-Gabriel Chiru suggested Movie Recommender, a system that uses the individual's known information to deliver movie recommendations without relying on previous searches. This system attempts to address the issue of unique recommendations that arises from neglecting data relevant to individual users. The psychological profile that the user has developed, their viewing and buffering history, and data from other websites including movie points and scores are all collected and taken into account. They are based on an estimate of collective similarity. The system is described as a hybrid model that recommends based on both content and collaborative filtering..

Chapter 3

Methodology

In various applications, several recommendation approaches have been presented and adopted. Hybrid technologies incorporate both the fundamental functions of a recommendation system. We give a quick summary of the most prominent recommendation/filtering approaches in RSs in this section.

3.1 Types of Recommendation system

3.1.1 Content-based recommender system (CBRS)

Many online movie song or commerce websites utilize a sort of recommendation system known as CBRS (content-based recommendation system). It is based on the user's information as well as similarities between things, services, or parts of content. The keywords, attributes, and qualities of the product in the database are used by CBRS to produce suggestions. Purchases, ratings (likes and dislikes), downloads, products searched for on a website and/or put to a basket, and product link clicks are all used to create the user profile.

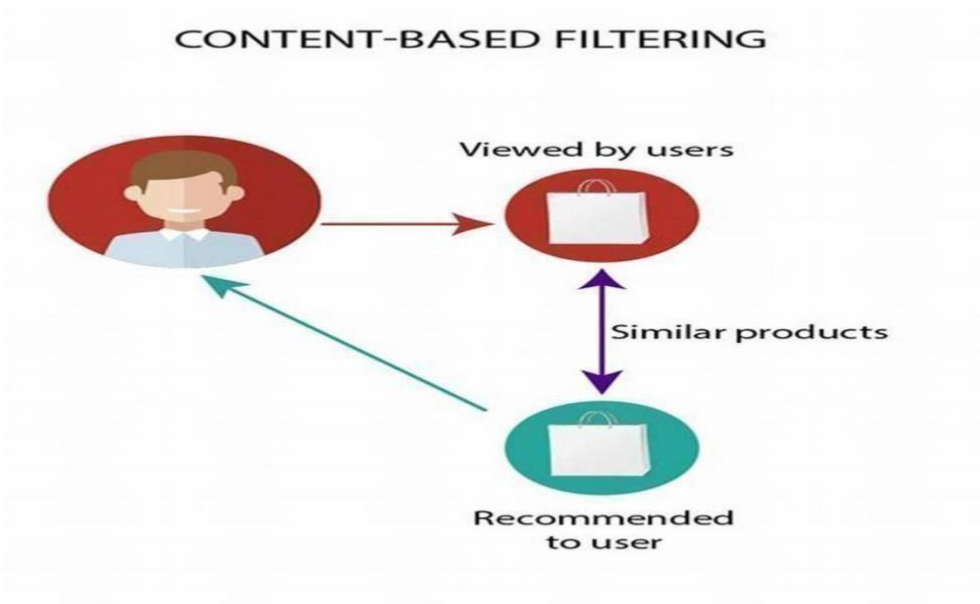


Figure 3.1: Content-based recommender system

Why use CBRS-

-Advantages of employing a content-based recommendation system include:

1-Not having to rely on other people's data to begin producing recommendations. CBSR begins producing relevant recommendations once a user browses, searches, or makes a purchase. This makes it suitable for small businesses that don't have access to a large amount of user data.

2-Suggestions are highly relevant to the user -Because the method relies on matching the characteristics or attributes of a database object with the user's profile and past similarity, content-based recommenders can be highly tailored according to the user's interests, including recommendations for new items.

3-Suggestions are transparent to the user-Highly relevant.

3.1.2 Collaborative filtering recommender system (CFRS)

The collaborative filtering recommender system is the most widely used system for filtering content based on user engagement and data collected from various users. CFRS adheres to the adage that "a man is known by the company he maintains." The majority of collaborative filtering techniques are based on similarity indexes. For example, if user1's and user2's purchase histories are highly similar, it's likely that if user1 buys a product, user2 will buy the same or comparable product. CF methods maintain note of a user's previous reviews and ratings on things so that comparable items can be recommended in the future. Even if the user has never dealt with a particular item, if his peers have, it would be recommended to him.

Figure 3.2: Collaborative -based recommender system

Advantages of CFRS- CFRS benefits include: -no need to grasp item content -no item cold start problem - we can anticipate item rating even when no information on the item is provided.
- track the evolution of user interest over time

Typical workflow of collaborative system -

Typical collaboration system work flow -

1. The first user expresses his or her choices by assigning a score to each item. These scores might be interpreted as a rough estimate of the user's interest in the related domain.
2. The system compares this user's ratings to those of other users to determine who has the most "similar"tastes.

3.1.3 Hybrid recommender system (HRS)

Hybrid RS is the result of combining different filtering algorithms, as the name implies.CBS and CFRS are the most prevalent HRS combo. The goal of merging several filtering algorithms is to increase recommendation accuracy by smoothing out the shortcomings of CBRS and CFRS, as well as to improve performance.

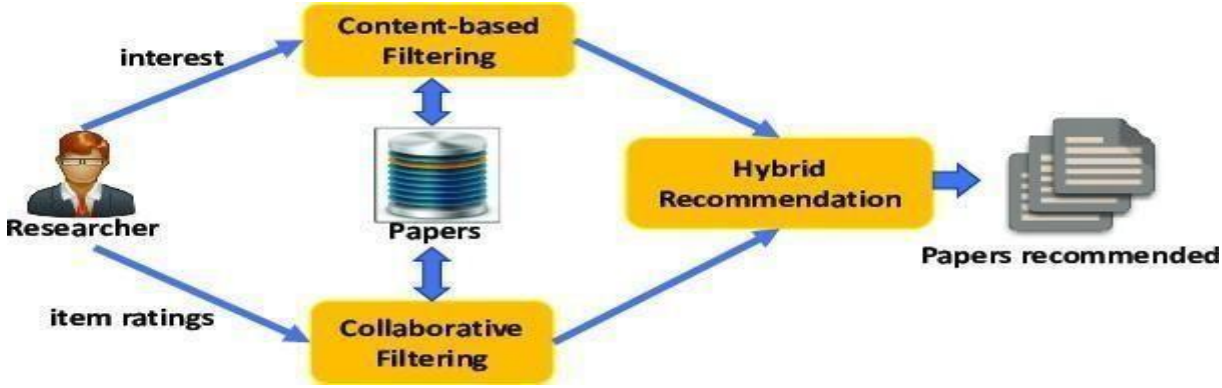
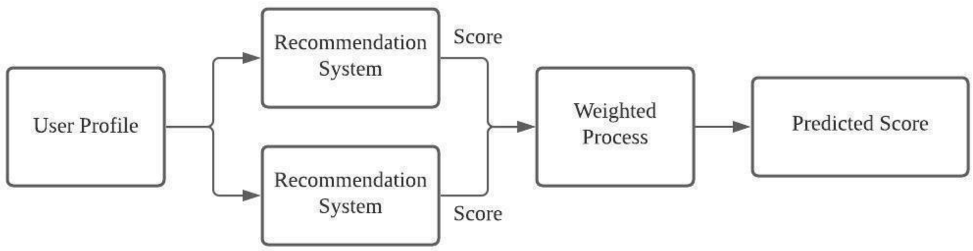


Figure 3.3: **Hybrid recommender system (HRS)**

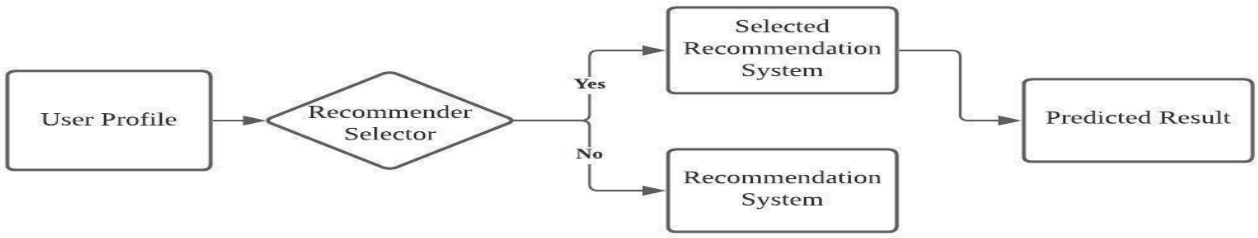
Types of Hybrid system –

1- Weighted



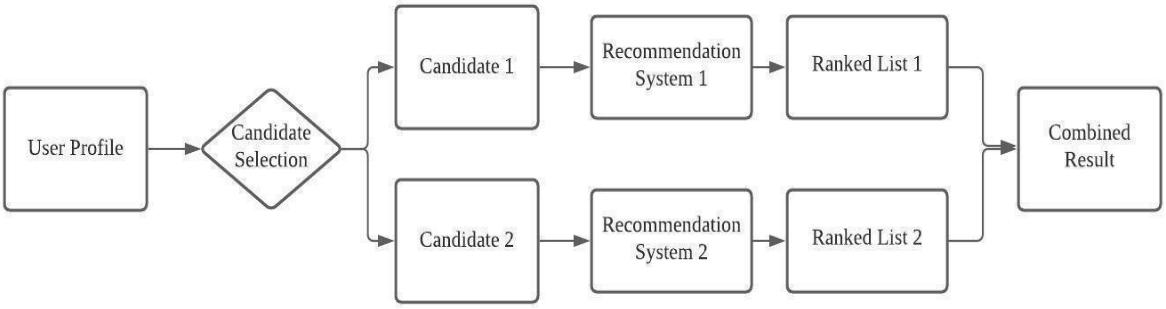
The weighted recommendation system uses output of different model combine the result using static weighting , which the does not change across the train and test set.

2- Switching -



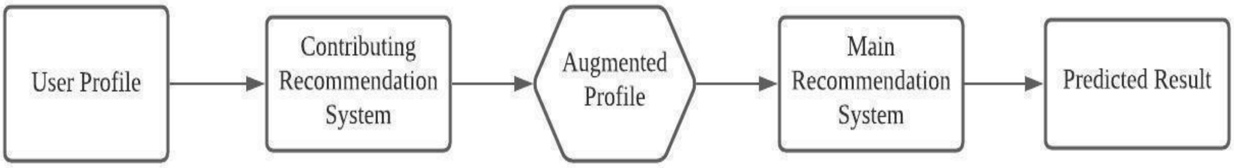
The switching hybrid selects a single recommendation system based on the situation. The model is used to create the item-level sensitive data-set. The switching hybrid technique adds an additional layer to the recommendation model that selects the optimal model to use.

3- Mixed -



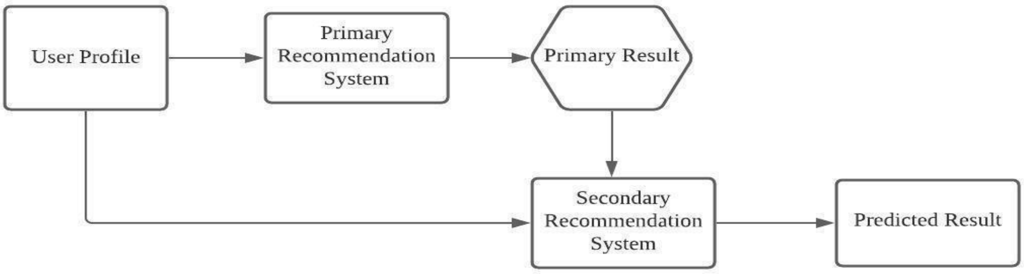
A mixed hybrid technique generates a variety of candidates based on the user's profile and features. The system feeds a variety of user data into the model, which then combines the results to come up with a recommendation.

4- Feature augmentation



This is a contributing model for providing a rating or classifying a user/item profile, and it will be utilised in the primary recommendation system to generate results. This suggestion system has the ability to increase the core system's accuracy and performance.

5- Cascade



Cascade hybrid system employs a rigorous hierarchical structure system, with primary results generated by the main recommendation system and minor issues resolved by the secondary model.

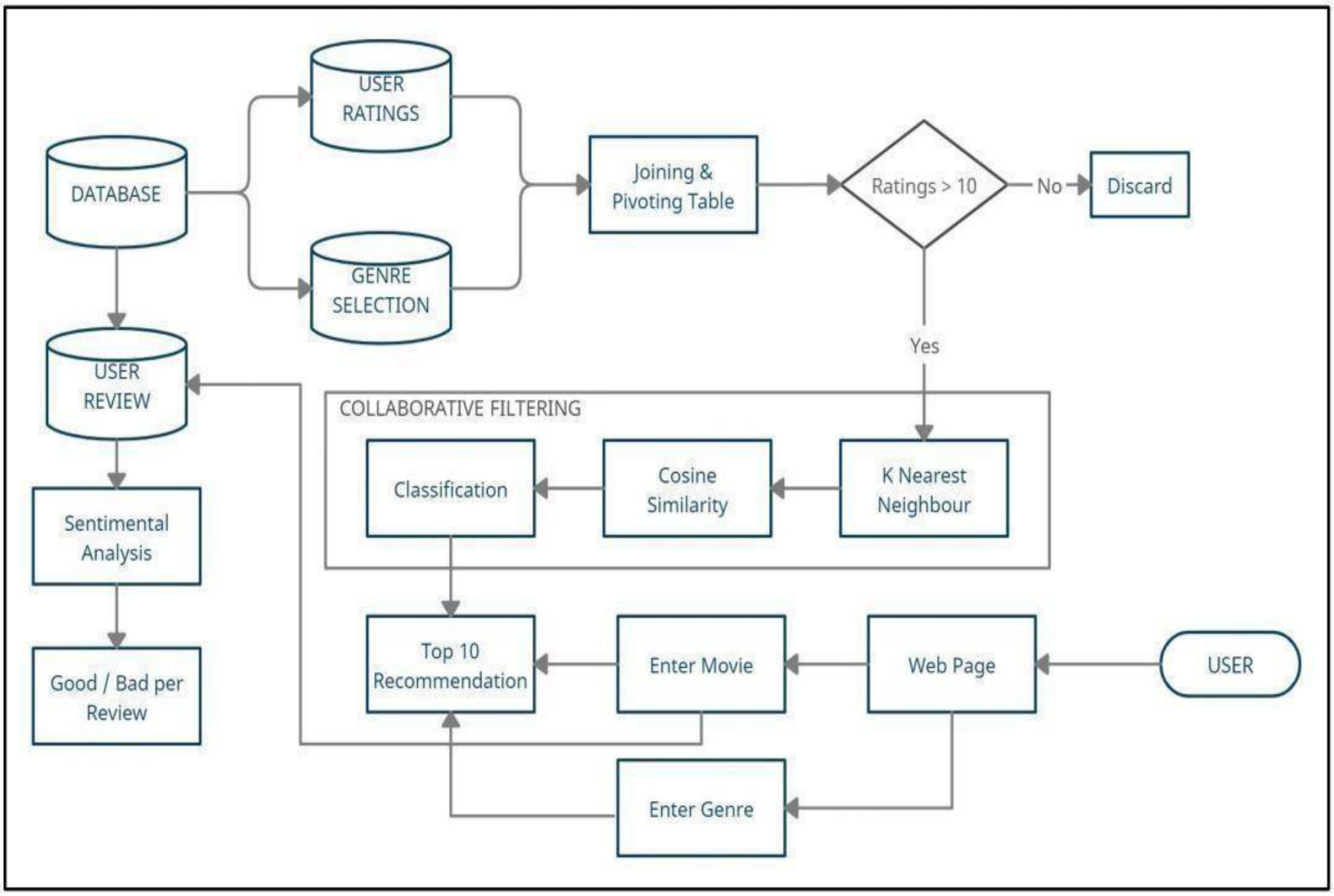


Figure 3.4: Block Diagram of Movie Recommendation System Model

Broadly, the whole process of classification can be divides into 2 phases:

- **Phase 1**: Create a movie recommendation system using Collaborative Filtering andmachine learning algorithms such as K Nearest Neighbours.
- **Phase 2**: Apply sentiment analysis to categorize user comments on a particular movie.
- **Phase 3**: Additional Content Based Filtering is performed using Neural Network toperform Matrix Factorization.

3.2 K-Nearest-Neighbours (KNN)

The K-Nearest-Neighbors (KNN) formula is that the most elementary non-parametric classification technique. it's supported supervised learning, which suggests it makes no assumptions concerning the fundamental data set. it's well-known for its simple use and effectiveness. the info points are classified based on their similarity in a very labeled coaching data set, permitting the category of unlabelled data to be predicted.

It' conjointly referred to as the lazy learner rule as a result of it can't learn from the coaching set right away. it's accustomed classify information in a very specific region supported the nearest or close training samples. This technique is utilized because it is easy to implement and takes a brief quantity of time to compute. It calculates its nearest neighbours victimisation the euclidian distance for continuous data.

For a brand new input, the K closest neighbours are determined, and therefore the majority of the neighbouring data determines the new input' classification. Despite the actual fact that this classifier is basic, the worth of 'K' is crucial in characteristic the unlabelled information. There are many strategies for determinative the values for 'K,' but we are able to simply run the classifier multiple times with totally different values to see that one produces the most effective results. as a result of all of the computations are done once the coaching data is classified, instead of when it's met within the dataset, the computation value is considerably higher.

It' a lazy learning rule because it doesn't do something except store the training data and memorize the data set when it' being trained. The coaching data set isn't accustomed conduct generalisation. As a result, throughout the testing stage, the whole basic datas et being trained is required.KNN forecasts continuous values in regression. This range is that the average of its K nearest neighbours' numbers. The KNN technique is employed to classify data..

Classification is divided into two steps:
1. Building a Classifier: A classifier is built using the training data.
2. The classifier's performance.

As per the closest neighboring technique, the new untagged sample is to classified by crucial that categories its neighbouring sample belong to. KNN algorithmic rule utilizes this idea in its calculation. just in case of KNN algorithm, a specific worth of K is mounted which helps U.S. in classifying the unknown tuple. once a replacement unlabeled tuple is encountered within the data-set, KNN performs two operations:

First, it analyzes the K points closest to the new information point, i.e. , the K nearest neighbours.
Second, victimisation the neighbours' categoryes, KNN determines on that class ought to the new data be classified into.

Once new data is added, it's mechanically classified. It' a lot of helpful in a very dataset that' roughly separated into clusters and corresponds to a selected plot region. As a result, this algorithmic rule improves the accuracy of separating data inputs into separate categories in a more logical manner.

KNN determines which class has the best variety of points sharing the shortest distance from the data point to be categorised. As a result, the geometer distance between the check sample and the provided coaching samples should be determined. we tend to simply take the bulk of K-Nearest Neighbors to predict the training example' category when we've gathered K-Nearest Neighbors.The worth of K, the euclidean distance, and also the normalization of the parameters are all aspects that influence the performance of KNN. The stages to grasp the algorithm' elaborated operation are as follows:

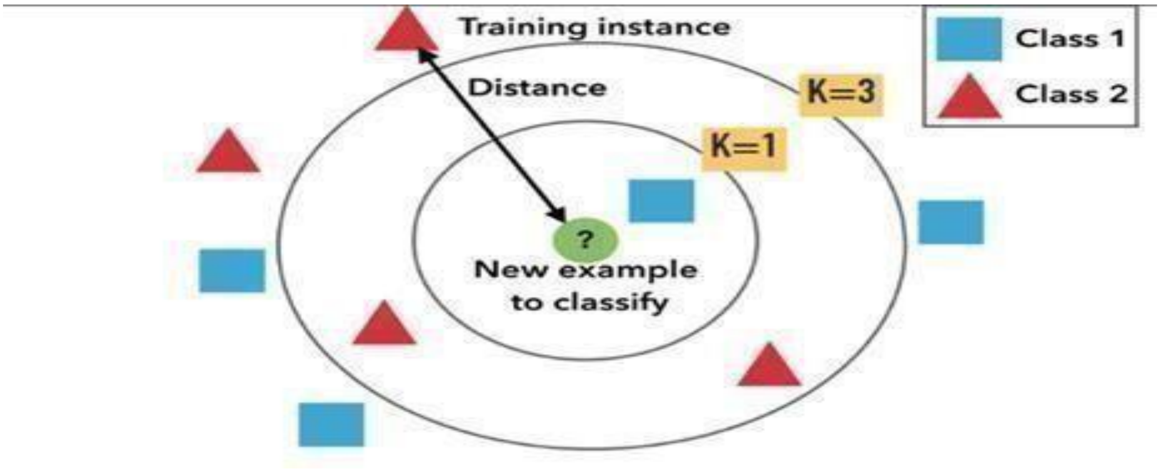


Figure 3.5: K's value Prediction

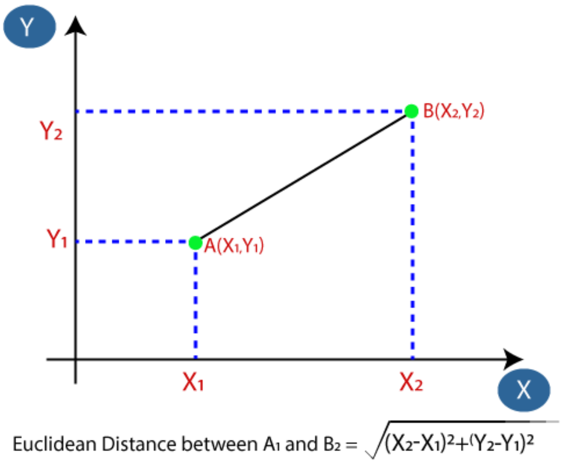
Given the training dataset : { (x(1) , y(1)) , (x(2) , y(2)), …… , (x(m) , y(m)) }

Step1: Store the training set

Step2: For each new unlabeled data,

1. Calculate euclidean distance with all training data points using the formula

   $\sqrt{(x1 - x2)^2 + (y1 - y2)^2}$



2. Find the k- nearest neighbours

3. Assign class containing the maximum number of nearest neighbours.

Following the storage of the training, all parameters should be normalised to form computations easier. The classification outcome is littered with the worth of 'K.' the quantity of neighbors that must be thought of is decided by the input value 'K.' the worth of 'K' has a control on the formula since it permits America to outline the borders of every class.

TO DETERMINE K:

By initially reviewing the data, the best value of K is determined.
Larger K values are more exact since they lower net noise, however this is not always the case.
Cross validation can also be used to find a good K value.

When K=1, the data is simply assigned to the nearest neighbor's class. The error rate for the

training knowledge is incessantly zero once K=1. this happens as a result of the closest point to any coaching information is that the training data point itself. As a result, the most effective outcomes are obtained when K=1.
The bounds, however, are over-fitted when K=1.
The formula is too sensitive to noise when 'k' is incredibly tiny. The training and validation sets should be separated from the initial dataset so as to get a favourable K value. the end result is unclear if the 2 Nearest Neighbors (K=2) belong to 2 separate classes. As a result, we tend to raise the number of nighest neighbours to a better figure (say 5-nearest neighbours).
Larger values of 'K' swish down the category boundaries, that isn't continually fascinating as a result of alternative classes' points might find yourself within the neighbourhood. It' troublesome to work out the worth of K once the coaching knowledge points are strewn about.

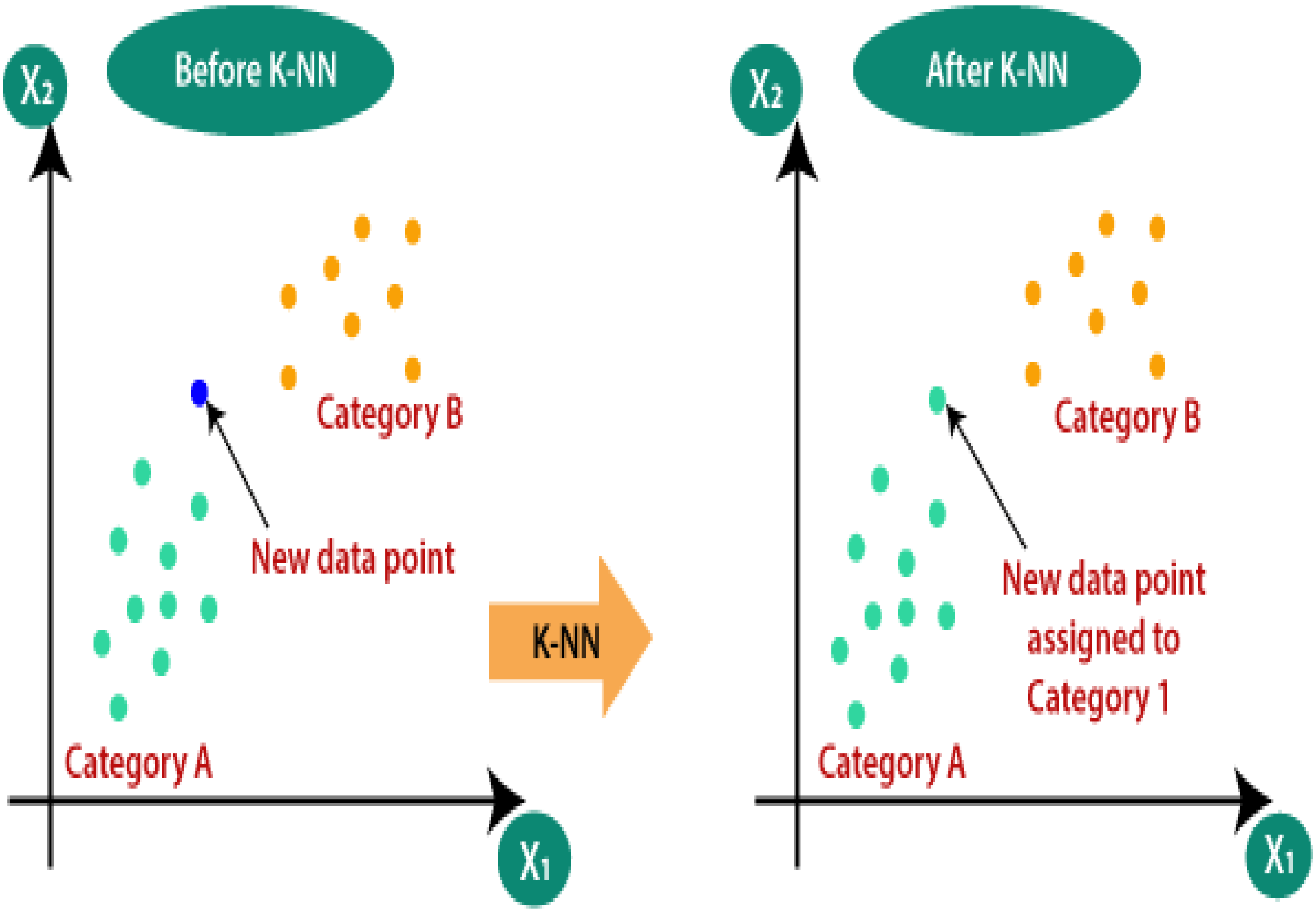


Figure 3.6: Before and After K-NN

3.3 Cosine Similarities

The similarity of two vectors in an inner product space is measured by cosine similarity. It determines whether two vectors are pointing in the same general direction by measuring the cosine of the angle between them. In text analysis, it's frequently used to determine document similarity..

Thousands of characteristics can be used to characterize a document, each of which records the frequency of a specific word (such as a keyword) or phrase in the document. As a result, each document is an object that is represented by a term-frequency vector. Figure 3.7 shows that Document1 has five instances of the phrase team and three instances of the word hockey.A count value of 0 indicates that the term coach does not appear anywhere in the document. Such information can be quite asymmetric..

Document	*team*	*coach*	*hockey*	*baseball*	*soccer*	*penalty*	*score*	*win*	*loss*	*seaso*
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

Figure 3.7

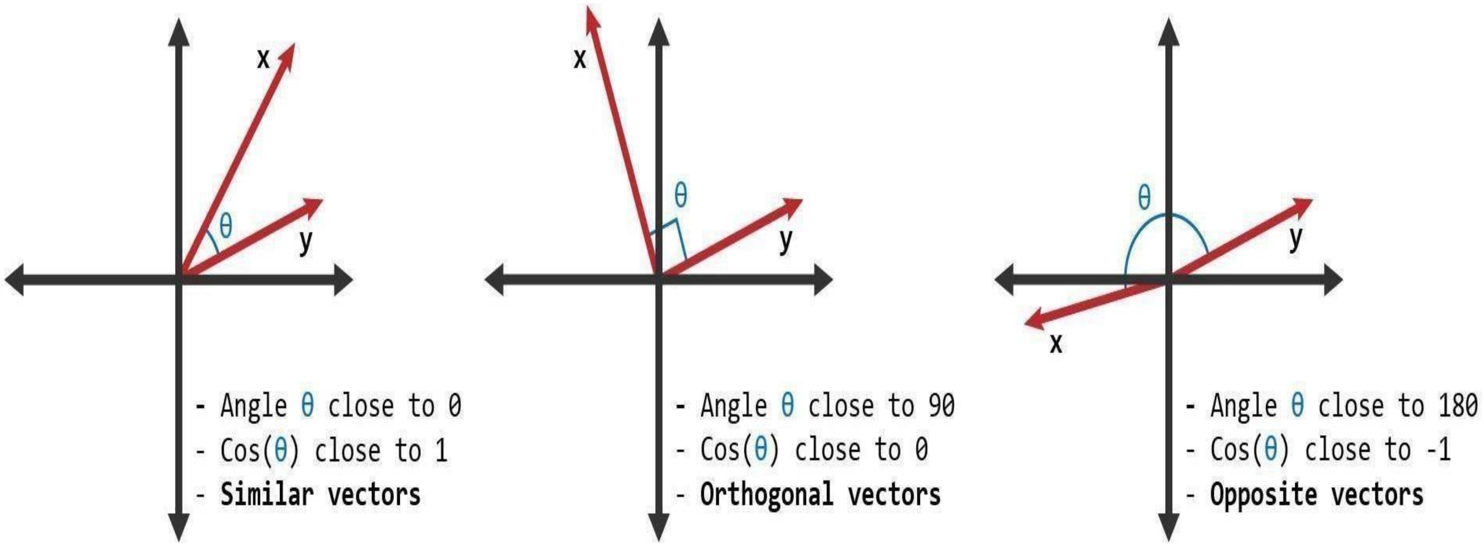


Figure3.8

Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words. Let ***v*** and ***w*** be two vectors for comparison. Using the cosine measure as a similarity function, we have

$$\text{COSINE SIMILARITY(A,B)} = \frac{A.B}{|A||B|} = \sum_{i=0}^{n} \frac{Ai*Bi}{\sqrt{\sum Bi^2}\sqrt{\sum Ai^2}}$$

Advantages :

- The cosine similarity is advantageous because, despite the fact that the two identical data objects are separated by the Euclidean distance due to their size, they may have a smaller angle between them.
- When plotted on a multi-dimensional space, the cosine similarity captures the orientation (angle) of the data objects rather than their magnitude.

3.4 Sentimental Analysis

Sentiment is a feeling-driven attitude, idea, or judgement. Sentiment analysis, often known as opinion mining, is the study of people's attitudes toward specific entities. When it comes to sentiment data, the internet is a valuable resource. People can upload their own content on various social media platforms, such as forums, microblogs, and online social networking sites, from the perspective of the user. From the standpoint of a researcher, many social mediasites expose their application programming interfaces (APIs), allowing researchers and developers to collect and analyse data. For example, Twitter now offers three APIs: the RESTAPI, the Search API, and the Streaming API..

Developers can also combine APIs to build their own apps. As a result, sentiment analysis looks to be built on a solid basis based on massive amounts of web data. However, there are a number of drawbacks to using this type of internet data for sentiment research. The first flaw is that the quality of people's opinions cannot be guaranteed because they can publish whatever they wish.

The technique of identifying whether a text contains negative, positive, or neutral emotions is referred to as "sentiment analysis." This type of text analytics employs natural language processing (NLP) and machine learning. Sentiment analysis is also known as "opinion mining" or "emotion artificial intelligence."

Sentiment is a highly subjective concept. Tone, context, and language are all used by humans to express meaning. Our understanding of that meaning is influenced by our personal experiences and unconscious prejudices. Let's take a look at a customer review for a new SaaS product to see how this works::

“Gets the job done, but it’s not cheap!”

This statement contains both negative and good sentiment. The price is linked to negative mood.

The product's functionality is linked to positive sentiment. But what isthe sentence's overall meaning?

Human prejudice and error can enter in at this point. Because the reviewer emphasizes functioning in a favourable attitude, human analysts may consider this statement to be positive overall. They may, on the other hand, focus on the unfavourable statement about price and label it as such. This is only one example of how subjectivity may affecthow people feel about things.

To generate more accurate insights, sentiment analysis tools use uniform criteria. A machine learning model, for example, can be trained to recognise two aspects with two different feelings. It would keep track of the details while averaging the overallfeeling as neutral.

3.5 Neural Network

ANNs are synthetic adaptive structures which can be stimulated through the human brain's functioning processes.

They are structures that may extrade their inner shape in reaction to a feature goal. They are particularly well-proper to managing nonlinear issues, as they are able to recreate the bushy policies that control the great answer for those situations. The nodes, additionally referred to as processing elements (PE), and the connections shape the muse of the ANN. Each node has its personal enter, which it makes use of to receive communications from different nodes and/or the environment, as well as an output, which it makes use of to engage with different nodes or the environment. Finally, every node has a feature that turns its

The whole ANN starts to research due to this dynamic. The 'Law of Learning' describes the procedure via which nodes adapt themselves. An ANN's overall dynamic is time-dependent. In reality, the surroundings should act at the ANN a couple of instances for the ANN to regulate its personal connections. The ANN's surroundings is made from data. As a result, one of the key mechanisms that characterise ANNs, which might be taken into consideration adaptive processing systems, is the studying procedure.

.

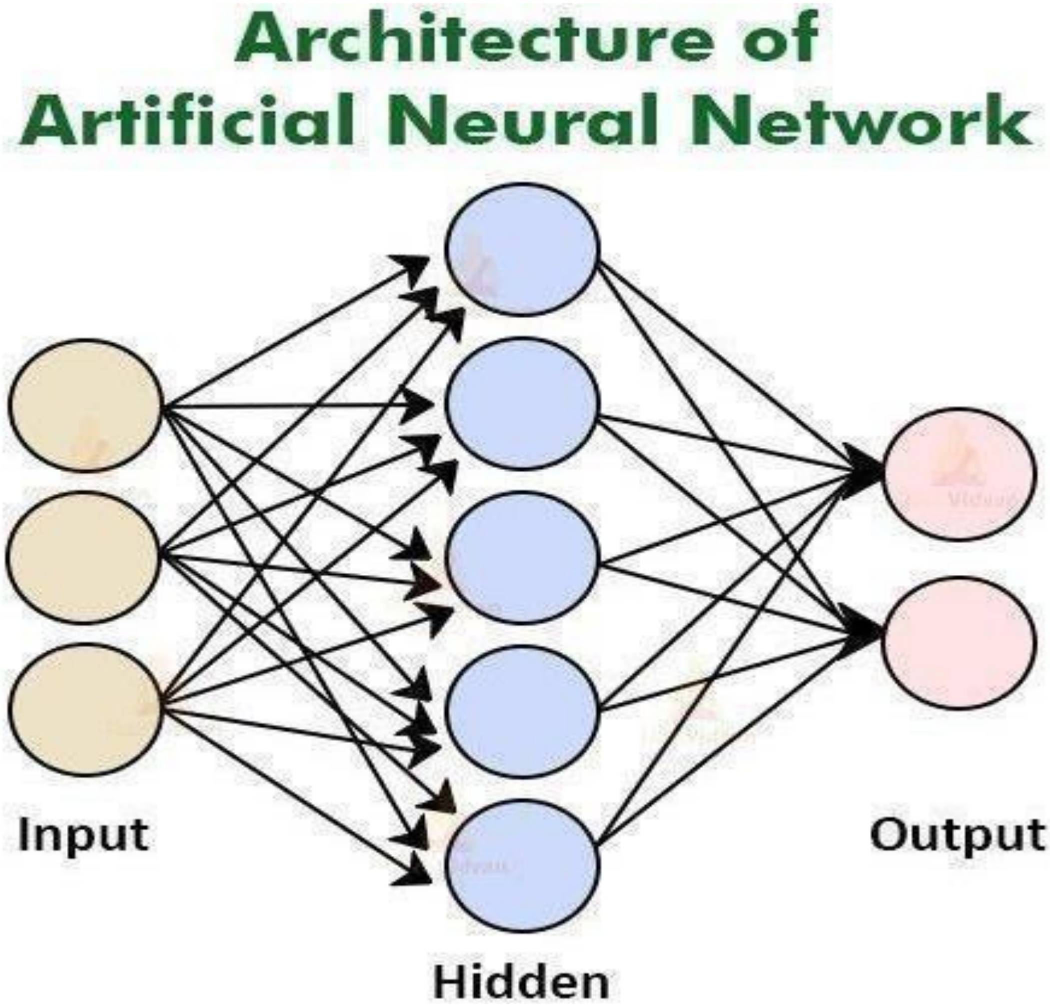


Figure 3.9: Neural Networks

The learning process is one way to adapt the connections of an ANN to the data structure that make up the environment and, therefore, a way to 'understand' the environment and the relationsthat characterize it. Neurons can be organized in any topological manner (e.g. one- or two- dimensional layers, three-dimensional blocks or more-dimensional structures), depending on the quality and amount of input data. The most common ANNs are composed in a so-called feed forward topology. A certain number of PEs is combined to an input layer, normally depending onthe amount of input variables. The information is forwarded to one or more hidden layers working within the ANN.

The output layer, as the last element of this structure, provides the result. The output layer contains only one PE, whether the result is a binary value or a single number. Figure 2 represents the most popular architecture of neural networks: back propagation .

The term 'neural' comes from the fundamental purposeful unit of the human (animal) frightened system, the 'neuron' or nerve cells discovered within side the mind and different areas of the human (animal) body. A neural community is a group of algorithms that authenticate the underlying courting in a batch of information within side the equal manner because the human mind does. The neural community assists in converting the input in order that the community can produce the premier end result while not having to rewrite the output technique..

Parts of Neuron and their Functions

The typical nerve cell of the human brain comprises of four parts -

Function of Dendrite

It receives signals fromother neurons.

Soma (cell body)

It sums all the incoming signals to generate input.

Axon Structure

When the sum reaches a threshold value, the neuron fires, and the signal travels down the axon to the other neurons.

Synapses Working

The point of interconnection of one neuron with other neurons. The amount of signal transmitted depend upon the strength (synaptic weights) of the connections.

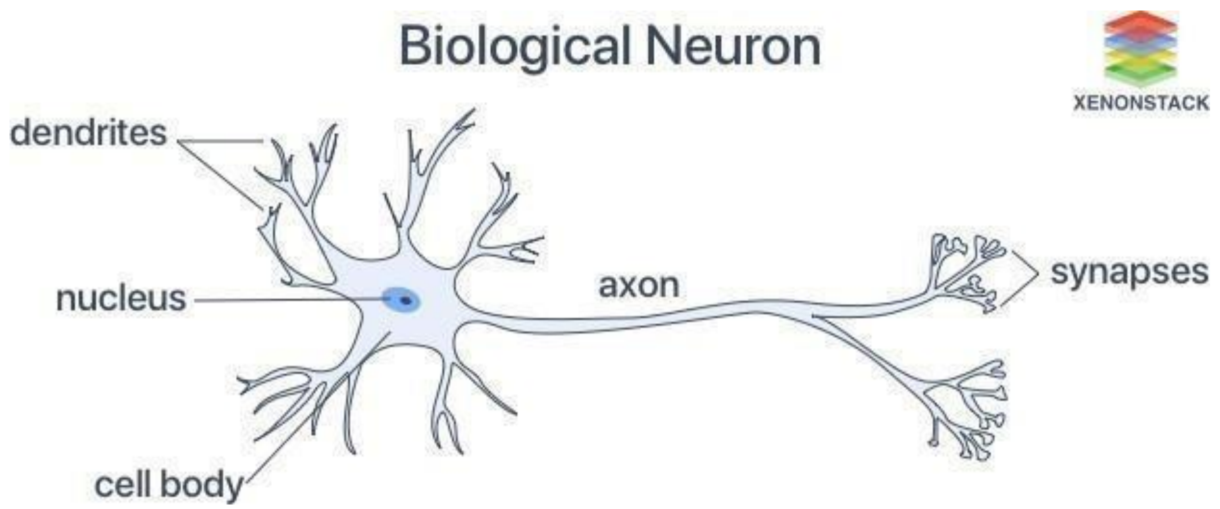


Figure 3.10: Structure of Neuron

Inhibitory (decreasing strength) or excitatory (increasing strength) connections can exist. In general, a neural network consists of a connected network of billions of neurons with trillions of interconnections.

Neural Networks resemble the human brain in the following two ways -

- 1-A neural network acquires knowledge through learning.
- 2- A neural network's knowledge is a store within inter-neuron connection strengths known as synaptic weights.

How Does Artificial Neural Network Works?

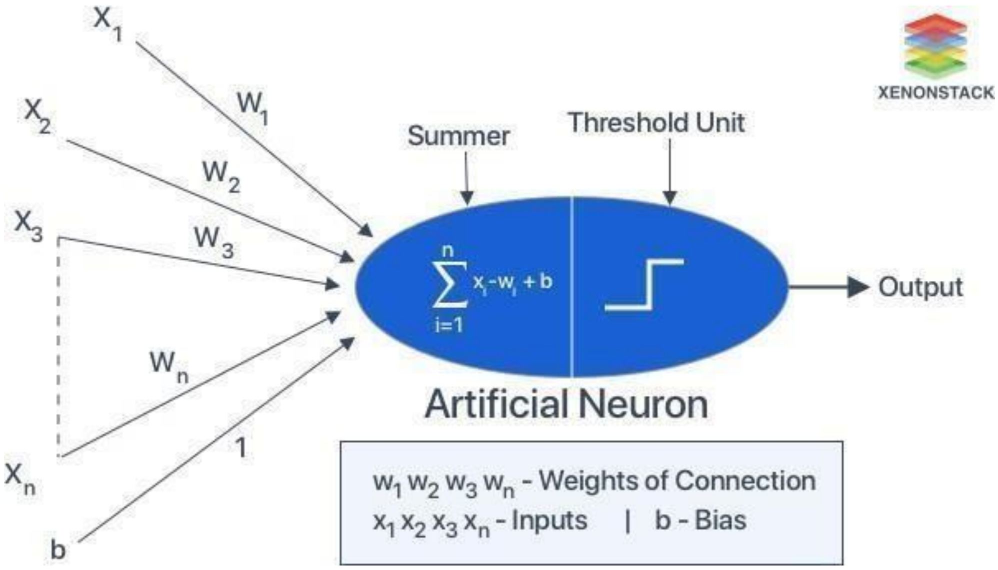


Figure 3.11: Working of Neural Network

Artificial Neural Networks can be thought of as weighted directed graphs with nodes representing artificial

neurons and directed edges with weights representing connections between neuron outputs and neuron inputs.

1. The Artificial Neural Network receives data from the outside environment in the form of patterns and vector images. The notation x(n) for n number of inputs designates these inputs. Each input is multiplied by the weights assigned to it. The information that the neural network uses to solve a problem is called weights. The strength of the connections between neurons within the Neural Network is typically represented by weight.
2. Inside the computing unit, all of the weighted inputs are added together (artificial neuron).
3. If the weighted total is 0, bias is applied to make the result non-zero or to increase the system responsescale. The weight and input in Bias are always set to '1'.
4. Any numerical number between 0 and infinity corresponds to the sum. The threshold value is set to limit the response so that it reaches the desired value. The sum is passed forward through an activation functionin this case. To get the desired result, the activation function is set to the transfer function. There are two types of activation functions: linear and nonlinear.

3.6 Matrix Factorization

In recommender systems, matrix factorization is a type of collaborative filtering technique. Theuser-item interaction matrix is decomposed into the product of two smaller dimensionality rectangular matrices by matric factorization procedures. Due to its success, this family of methodologies became well-known during the Netflix prize challenge, as recounted by SimonFunk in a 2006 blog post in which he shared his findings with the academic community. By giving varying regularisation weights to the latent components based on item popularity and user activity, the prediction outcomes can be improved.

Chapter 4

Implementation

1. Import required libraries

```
import pandas as pds
import numpy as np
import pickle
#import matrix_factorization_utilities
import scipy.sparse as sp
from scipy.sparse.linalg import svds
```

2. From the movie.csv file, extract genre and create column to put 0 and 1.

   0- If the particular movie does not have that particular genre
   1- If the particular movie had that particular genre

```
movies_with_genres.head()
```

	movieId	title	genres	Drama	Children	Adventure	War	Action	Thriller	Musical	...	IMAX	Crime	Animation	Mystery	Roma
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	0	1	1	0	0	0	0	...	0	0	1	0	
1	2	Jumanji (1995)	Adventure|Children|Fantasy	0	1	1	0	0	0	0	...	0	0	0	0	
2	3	Grumpier Old Men (1995)	Comedy|Romance	0	0	0	0	0	0	0	...	0	0	0	0	
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	1	0	0	0	0	0	0	...	0	0	0	0	
4	5	Father of the Bride Part II (1995)	Comedy	0	0	0	0	0	0	0	...	0	0	0	0	

5 rows × 23 columns

3. Finding mean and count of all the movieId whose reviews are more than 30.

```
#Get the average ratings of movies across all movies available
all_avg_rating = user_ratings['rating'].mean()
all_avg_rating
#set a minimum threshold for number of reviews that the movie has to have
min_reviews =30
min_reviews
movie_score = movie_avg_rating.loc[movie_avg_rating['count']>min_reviews]
movie_score.head()
```

movieId	mean	count
1	3.920930	215
2	3.431818	110
3	3.259615	52
5	3.071429	49
6	3.946078	102

4. Merging ratings mean and count with movie along with all the genre information. Also put a column of average weight that would be our future parameters for making our prediction.

```
#join movie details to movie ratings
movies_with_genres.index.name = None#.drop(columns=[])#, axis = 1)
movies_with_genres = movies_with_genres.rename_axis(None)
movie_score = pds.merge(movie_score,movies_with_genres,on='movieId')
#movie_score = movie_score.index.name = NULL
#join movie links to movie ratings
#movie_score = pd.merge(movie_score,links,on='movieId')
movie_score.head()
```

	movieId	mean	count	weighted_score	title	genres	Drama	Children	Adventure	War	...	IMAX	Crime	Animation	Mystery
0	1	3.920930	215	3.869578	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	0	1	1	0	...	0	0	1	0
1	2	3.431818	110	3.446762	Jumanji (1995)	Adventure|Children|Fantasy	0	1	1	0	...	0	0	0	0
2	3	3.259615	52	3.348131	Grumpier Old Men (1995)	Comedy|Romance	0	0	0	0	...	0	0	0	0
3	5	3.071429	49	3.234768	Father of the Bride Part II (1995)	Comedy	0	0	0	0	...	0	0	0	0

5. Creating a dataframe that is 2D matrix containing userId as Column and MovieId as row.
Having data of rating of each movie by each user in this 2D matrix and fill no rating movie with NAN.

```
# Creating a data frame that has user ratings accross all movies in form of matrix used in matrix factorisation
ratings_dataframe = pds.pivot_table(user_ratings, index='userId', columns='movieId', aggfunc=np.max)
```

```
#multiplication matrix
ratings_dataframe.head()
```

	rating																				
movieId	1	2	3	4	5	6	7	8	9	10	...	193565	193567	193571	193573	193579	193581	193583	193585	193587	193609
userId																					
1	4.0	NaN	4.0	NaN	NaN	4.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 9724 columns

6. Creating a pivot matrix.

```
#create a matrix table with movieIds on the rows and userIds in the columns.
#replace NAN values with 0
movie_without_NAN = filtered_ratings.pivot(index = 'movieId', columns = 'userId', values = 'rating').fillna(0)
movie_without_NAN.head()
```

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607	608	609	610
movieId																					
1	4.0	0.0	0.0	0.0	4.0	0.0	4.5	0.0	0.0	0.0	...	4.0	0.0	4.0	3.0	4.0	2.5	4.0	2.5	3.0	5.0
2	0.0	0.0	0.0	0.0	0.0	4.0	0.0	4.0	0.0	0.0	...	0.0	4.0	0.0	5.0	3.5	0.0	0.0	2.0	0.0	0.0
3	4.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
6	4.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	...	0.0	3.0	4.0	3.0	0.0	0.0	0.0	0.0	0.0	5.0

5 rows × 610 columns

7. Cosine similarity

```
[65] # Similarity of the movies based on the content
     cosine_similar
```

```
array([[5., 3., 1., ..., 0., 1., 1.],
       [3., 3., 0., ..., 0., 0., 0.],
       [1., 0., 2., ..., 0., 0., 1.],
       ...,
       [0., 0., 0., ..., 1., 0., 0.],
       [1., 0., 0., ..., 0., 2., 0.],
       [1., 0., 1., ..., 0., 0., 1.]])
```

8. Implementing collaborative and content filtering for the recommendation.

```
#Gets the other top 10 movies which are watched by the people who saw this particular movie
def get_other_movies(movie_name):
    #get all users who watched a specific movie
    df_movie_users_series = movie_ratings.loc[movie_ratings['title']==movie_name]['userId']
    #convert to a data frame
    df_movie_users = pds.DataFrame(df_movie_users_series,columns=['userId'])
    #get a list of all other movies watched by these users
    other_movies = pds.merge(df_movie_users,movie_ratings ,on='userId')
    #get a list of the most commonly watched movies by these other user
    other_users_watched = pds.DataFrame(other_movies.groupby('title')['userId'].count()).sort_values('userId',ascending=False)
    other_users_watched['perc_who_watched'] = round(other_users_watched['userId']*100/other_users_watched['userId'][0],1)
    return other_users_watched[:20]

    # Import linear_kernel
from sklearn.metrics.pairwise import linear_kernel

#specify model parameters
knn_model = NearestNeighbors(metric='cosine',algorithm='brute')
#fit model to the data set
knn_model.fit(movie_without_NAN)

# Compute the cosine similarity matrix
cosine_similar = linear_kernel(movie_content_df,movie_content_df)

#Gets the top 20 similar movies based on the content
def get_similar_movies_based_on_content(movie_index) :
    sim_scores = list(enumerate(cosine_similar[movie_index]))
    # 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[0:21]
    print(sim_scores)
    # Get the movie indices
    movie_indices = [j[0] for j in sim_scores]
    print(movie_indices)
    similar_movies = pds.DataFrame(movie_content_df_temp[['title','genres']].iloc[movie_indices])
    return similar_movies
```

9. Training and testing datasets

```
#import train_test_split module
from sklearn.model_selection import train_test_split
#take 80% as the training set and 20% as the test set
dataframe_train, dataframe_test= train_test_split(dataframe_with_index,test_size=0.2)
print(len(dataframe_train))
print(len(dataframe_test))

#Create two user-item matrices, one for training and another for testing
matrix_train_data = np.zeros((x_users, x_items))
    #for every line in the data
for line in dataframe_train.itertuples():
    #set the value in the column and row to
    #line[1] is userId, line[2] is movieId and line[3] is rating, line[4] is movie_index and line[5] is user_index
    matrix_train_data[line[5], line[4]] = line[3]
matrix_train_data.shape

#Create two user-item matrices, one for training and another for testing
matrix_test_data = np.zeros((x_users, x_items))
    #for every line in the data
for line in dataframe_test[:1].itertuples():
    #set the value in the column and row to
    #line[1] is userId, line[2] is movieId and line[3] is rating, line[4] is movie_index and line[5] is user_index
    #print(line[2])
    matrix_test_data[line[5], line[4]] = line[3]
    #train_data_matrix[line['movieId'], line['userId']] = line['rating']
matrix_test_data.shape
```

10. Neural Networks for Matrix multiplication

```
# Returns a neural network model which performs matrix factorisation
def matrix_factorisation_model_with_n_latent_factors(x_latent_factors) :
    movie_input = keras.layers.Input(shape=[1],name='Item')
    movie_embedding = keras.layers.Embedding(x_movies + 1, x_latent_factors, name='Movie-Embedding')(movie_input)
    movie_vec = keras.layers.Flatten(name='FlattenMovies')(movie_embedding)

    user_input = keras.layers.Input(shape=[1],name='User')
    user_vec = keras.layers.Flatten(name='FlattenUsers')(keras.layers.Embedding(x_users + 1, x_latent_factors,name='User-Emb

    prod = keras.layers.merge([movie_vec, user_vec], mode='dot',name='DotProduct')
    model = keras.Model([user_input, movie_input], prod)
    model.compile('adam', 'mean_squared_error')

    return model
```

11. Neural Network that does recommendation

```
# Returns a neural network model which does recommendation
def neural_network_model(n_latent_factors_user, n_latent_factors_movie):

    movie_input = keras.layers.Input(shape=[1],name='Item')
    movie_embedding = keras.layers.Embedding(n_movies + 1, n_latent_factors_movie, name='Movie-Embedding')(movie_input)
    movie_vec = keras.layers.Flatten(name='FlattenMovies')(movie_embedding)
    movie_vec = keras.layers.Dropout(0.2)(movie_vec)


    user_input = keras.layers.Input(shape=[1],name='User')
    user_vec = keras.layers.Flatten(name='FlattenUsers')(keras.layers.Embedding(n_users + 1, n_latent_factors_user,name='User-Embedding')(user_input)
    user_vec = keras.layers.Dropout(0.2)(user_vec)


    concat = keras.layers.merge([movie_vec, user_vec], mode='concat',name='Concat')
    concat_dropout = keras.layers.Dropout(0.2)(concat)
    dense = keras.layers.Dense(100,name='FullyConnected')(concat)
    dropout_1 = keras.layers.Dropout(0.2,name='Dropout')(dense)
    dense_2 = keras.layers.Dense(50,name='FullyConnected-1')(concat)
    dropout_2 = keras.layers.Dropout(0.2,name='Dropout')(dense_2)
    dense_3 = keras.layers.Dense(20,name='FullyConnected-2')(dense_2)
    dropout_3 = keras.layers.Dropout(0.2,name='Dropout')(dense_3)
    dense_4 = keras.layers.Dense(10,name='FullyConnected-3', activation='relu')(dense_3)


    result = keras.layers.Dense(1, activation='relu',name='Activation')(dense_4)
    adam = Adam(lr=0.005)
    model = keras.Model([user_input, movie_input], result)
    model.compile(optimizer=adam,loss= 'mean_absolute_error')
    return model
```

4.1 Experimental results

Getting movies based on Genre

```
best_movies_by_genre('Action',20)
```

	title	count	mean	weighted_s
474	Fight Club (1999)	218	4.272936	4.17
66	Star Wars: Episode IV - A New Hope (1977)	251	4.231076	4.15
227	Star Wars: Episode V - The Empire Strikes Back...	211	4.215640	4.12
429	Matrix, The (1999)	278	4.192446	4.12
229	Raiders of the Lost Ark (Indiana Jones and the...	200	4.207500	4.11
776	Dark Knight, The (2008)	149	4.238255	4.11
228	Princess Bride, The (1987)	142	4.232394	4.10
237	Apocalypse Now (1979)	107	4.219626	4.06
374	Saving Private Ryan (1998)	188	4.146277	4.05
238	Star Wars: Episode VI - Return of the Jedi (1983)	196	4.137755	4.05
668	Lord of the Rings: The Return of the King, The...	185	4.118919	4.03
132	Blade Runner (1982)	124	4.100806	3.98
795	Inglourious Basterds (2009)	88	4.136364	3.97
32	Braveheart (1995)	237	4.031646	3.97
815	Inception (2010)	143	4.066434	3.96

```
#run function to return top recommended movies by genre
best_movies_by_genre('Musical',20)
```

	title	count	mean	weighted_scor
98	Lion King, The (1994)	172	3.941860	3.8764
182	Singin' in the Rain (1952)	47	4.074468	3.8512
208	Willy Wonka & the Chocolate Factory (1971)	119	3.873950	3.7989
202	Sound of Music, The (1965)	64	3.937500	3.7983
190	My Fair Lady (1964)	35	4.042857	3.7930
191	Wizard of Oz, The (1939)	92	3.880435	3.7872
199	Mary Poppins (1964)	71	3.887324	3.7727
445	South Park: Bigger, Longer and Uncut (1999)	76	3.861842	3.7598
141	Aladdin (1992)	183	3.792350	3.7513
243	Blues Brothers, The (1980)	84	3.809524	3.7284
147	Beauty and the Beast (1991)	146	3.770548	3.7246
378	Jungle Book, The (1967)	53	3.830189	3.7114
358	Labyrinth (1986)	42	3.821429	3.6881
272	Fantasia (1940)	53	3.783019	3.6812
730	Walk the Line (2005)	39	3.756410	3.6456

Getting movies based on Cosine similarities

```
[ ] print_similar_movies(96079)
```

```
Recommendations for 7955    Skyfall (2012)
Name: title, dtype: object:

1: 6346    Casino Royale (2006)
Name: title, dtype: object, with distance of 0.3872801347983559:
2: 6886    Quantum of Solace (2008)
Name: title, dtype: object, with distance of 0.39021366042536676:
3: 8159    Star Trek Into Darkness (2013)
Name: title, dtype: object, with distance of 0.40121434472147643:
4: 7620    X-Men: First Class (2011)
Name: title, dtype: object, with distance of 0.410776231213382:
5: 7888    Prometheus (2012)
Name: title, dtype: object, with distance of 0.42116331218812264:
6: 7154    Zombieland (2009)
Name: title, dtype: object, with distance of 0.43147702340995964:
7: 7693    Avengers, The (2012)
Name: title, dtype: object, with distance of 0.43931631271892824:
8: 7768    Dark Knight Rises, The (2012)
Name: title, dtype: object, with distance of 0.4492845508571931:
9: 8681    Mad Max: Fury Road (2015)
Name: title, dtype: object, with distance of 0.46602536551026486:
10: 8683    Star Wars: Episode VII - The Force Awakens (2015)
Name: title, dtype: object, with distance of 0.4699041926734122:
```

Getting movies based on the content

```
[(0, 5.0), (559, 5.0), (1706, 5.0), (2250, 5.0), (2355, 5.0), (2809, 5.0), (3000, 5.0), (3194, 5.0), (3568, 5.0), (5490, 5.0), (5819, 5.0), (5977, 5.0), (6194, 5.0), (6260, 5.0), (6
[0, 559, 1706, 2250, 2355, 2809, 3000, 3194, 3568, 5490, 5819, 5977, 6194, 6260, 6448, 6486, 6626, 6948, 7355, 7360, 7530]
```

	title	genres
0	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
559	Space Jam (1996)	Adventure|Animation|Children|Comedy|Fantasy|Sc...
1706	Antz (1998)	Adventure|Animation|Children|Comedy|Fantasy
2250	Who Framed Roger Rabbit? (1988)	Adventure|Animation|Children|Comedy|Crime|Fant...
2355	Toy Story 2 (1999)	Adventure|Animation|Children|Comedy|Fantasy
2809	Adventures of Rocky and Bullwinkle, The (2000)	Adventure|Animation|Children|Comedy|Fantasy
3000	Emperor's New Groove, The (2000)	Adventure|Animation|Children|Comedy|Fantasy
3194	Shrek (2001)	Adventure|Animation|Children|Comedy|Fantasy|Ro...
3568	Monsters, Inc. (2001)	Adventure|Animation|Children|Comedy|Fantasy
5490	Twelve Tasks of Asterix, The (Les douze travau...	Action|Adventure|Animation|Children|Comedy|Fan...
5819	Robots (2005)	Adventure|Animation|Children|Comedy|Fantasy|Sc...
5977	Valiant (2005)	Adventure|Animation|Children|Comedy|Fantasy|War
6194	Wild, The (2006)	Adventure|Animation|Children|Comedy|Fantasy
6260	Ant Bully, The (2006)	Adventure|Animation|Children|Comedy|Fantasy|IMAX

Getting movies based on user profile, what type of movies user like and ratings given by user.

```
Top 10 predictions for User 1
```

	ratings	movieId	title	genres
0	5.497649	2670	Run Silent Run Deep (1958)	War
1	5.432393	910	Some Like It Hot (1959)	Comedy|Crime
2	5.371723	1754	Fallen (1998)	Crime|Drama|Fantasy|Thriller
3	5.269211	835	Foxfire (1996)	Drama
4	5.233078	551	Nightmare Before Christmas, The (1993)	Animation|Children|Fantasy|Musical
5	5.232338	3668	Romeo and Juliet (1968)	Drama|Romance
6	5.231938	46	How to Make an American Quilt (1995)	Drama|Romance
7	5.229878	898	Philadelphia Story, The (1940)	Comedy|Drama|Romance
8	5.196545	2019	Seven Samurai (Shichinin no samurai) (1954)	Action|Adventure|Drama
9	5.194960	968	Night of the Living Dead (1968)	Horror|Sci-Fi|Thriller

```
Top 10 predictions for User 10
```

	ratings	movieId	title	genres
0	4.685765	3189	My Dog Skip (1999)	Children|Drama
1	4.661531	7449	Godsend (2004)	Drama|Horror|Thriller
2	4.253596	2670	Run Silent Run Deep (1958)	War
3	4.179119	7022	Battle Royale (Batoru rowaiaru) (2000)	Action|Drama|Horror|Thriller
4	4.122558	4131	Making Mr. Right (1987)	Comedy|Romance|Sci-Fi
5	4.072820	3635	Spy Who Loved Me, The (1977)	Action|Adventure|Thriller
6	3.986923	3633	On Her Majesty's Secret Service (1969)	Action|Adventure|Romance|Thriller
7	3.968859	5901	Empire (2002)	Crime|Drama
8	3.908728	694	Substitute, The (1996)	Action|Crime|Drama
9	3.470427	5780	Polyester (1981)	Comedy

4.2 Comparison with previous model

Previous model is recommending movie on the basis of content only(lets say same genre types movies are recommended to user.

	title	genres
0	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
559	Space Jam (1996)	Adventure|Animation|Children|Comedy|Fantasy|Sc...
1706	Antz (1998)	Adventure|Animation|Children|Comedy|Fantasy
2250	Who Framed Roger Rabbit? (1988)	Adventure|Animation|Children|Comedy|Crime|Fant...
2355	Toy Story 2 (1999)	Adventure|Animation|Children|Comedy|Fantasy
2809	Adventures of Rocky and Bullwinkle, The (2000)	Adventure|Animation|Children|Comedy|Fantasy
3000	Emperor's New Groove, The (2000)	Adventure|Animation|Children|Comedy|Fantasy
3194	Shrek (2001)	Adventure|Animation|Children|Comedy|Fantasy|Ro...
3568	Monsters, Inc. (2001)	Adventure|Animation|Children|Comedy|Fantasy
5490	Twelve Tasks of Asterix, The (Les douze travau...	Action|Adventure|Animation|Children|Comedy|Fan...
5819	Robots (2005)	Adventure|Animation|Children|Comedy|Fantasy|Sc...
5977	Valiant (2005)	Adventure|Animation|Children|Comedy|Fantasy|War
6194	Wild, The (2006)	Adventure|Animation|Children|Comedy|Fantasy
6260	Ant Bully, The (2006)	Adventure|Animation|Children|Comedy|Fantasy|IMAX

New model is taking various parameters into consideration

1. Taken Rating database into account that has ratings done by different user for different movies.(collaborative Based filtering).
2. Individual profile is taken into consideration. Like what type of movies are rated by that particular Userid
   (content based filtering)
3. Model then merges both the recommended movies and present to user.

```
[ ]  print_similar_movies(96079)
```

```
Recommendations for 7955    Skyfall (2012)
Name: title, dtype: object:

1: 6346    Casino Royale (2006)
Name: title, dtype: object, with distance of 0.3872801347983559:
2: 6886    Quantum of Solace (2008)
Name: title, dtype: object, with distance of 0.39021366042536676:
3: 8159    Star Trek Into Darkness (2013)
Name: title, dtype: object, with distance of 0.40121434472147643:
4: 7620    X-Men: First Class (2011)
Name: title, dtype: object, with distance of 0.410776231213382:
5: 7888    Prometheus (2012)
Name: title, dtype: object, with distance of 0.42116331218812264:
6: 7154    Zombieland (2009)
Name: title, dtype: object, with distance of 0.43147702340995964:
7: 7693    Avengers, The (2012)
Name: title, dtype: object, with distance of 0.43931631271892824:
8: 7768    Dark Knight Rises, The (2012)
Name: title, dtype: object, with distance of 0.4492845508571931:
9: 8681    Mad Max: Fury Road (2015)
Name: title, dtype: object, with distance of 0.46602536551026486:
10: 8683    Star Wars: Episode VII - The Force Awakens (2015)
Name: title, dtype: object, with distance of 0.4699041926734122:
```

Chapter 5

Conclusions

5.1 Conclusions

This document is largely divided in sub gadget . First one specializes in the film advice component and the second specializes in the sentimental evaluation component. For film advice component , we makes use of cosine similarity which makes use of angular area among the vectors to locate the similarity rating which are associated with film entered via way of means of the consumer primarily based totally on various factors inclusive of style of the film , cast, administrators etc,.Cosine Similarity proved to be pretty correct in recommending the film.

Sentimental evaluation performs a crucial position within side the study. It pursuits to categorize the consumer's assessment as fine and negative. The algorithms used for sentimental evaluation are Naives bayes and Support Vector Machine .We makes use of algorithms to locate which one is the first-class to categorise the assessment due to big range and complexity of assessment dataset.

SVM proved to be extra correct than NB.

Although each the gadget works appropriately however nonetheless have a few limitation.First one is that if the film entered via way of means of the consumer isn't gift withinside the information set, or consumer entered incorrect film name , then the gadget could be failed . Second one is concerning linguistic barrier in doing sentimental evaluation, due to the fact until now best opinions which are written in English are best recognized. The sentimental evaluation will offer incorrect rationalization if the assessment is written in ironic or abusive language..

5.2 Future Works

In future , for additional promotion and building of moving-picture show culture community, we'll use intelligent machine learning system to analyse the movie and movie review provided by the user thus on decide the sort of movie user like and topics they're curious about order to attain more correct recommendation results .

Neural Networks and Deep Learning are all the fad during a type of sectors in recent years, and it's that they'll even be accustomed solve difficulties with recommendation systems..

The capability to infer latent properties is one in every of the most options of Deep Learning, that is analogous to matrix factoring. Deep Learning, on the opposite hand, will make amends for a number of matrix factorization' flaws, equivalent to the lack to incorporate time within the model – one thing that normal matrix factorization isn't designed to do. Deep Learning, on the other hand, can create use of perennial Neural Networks, which are created expressly for time and sequence data.

Because there are usually discriminatory seasonal effects, it' vital to include time into a recommender system. In December, for example, it's expected that a lot of people are going to be viewing Christmas movies and buying vacation decorations.

mountain Allison conjointly mentioned the importance of determinative what would happen if a client was shown a sub-optimal recommendation. as a result of the aim during this state of affairs is to gift purchasers a recommendation and so record what they do, this can be a reinforcement learning strategy. Customers could also be offered one thing that doesn't seem to be the best choice solely to observe however they react, which is able to increase longterm learning..

Recommender systems is a really effective tool in an exceedingly company' armoury, and future advances will solely increase its worth. a number of the uses embrace the power to predict seasonal purchases supported suggestions, determine vital transactions, and supply higher recommendations to clients, all of which can facilitate enhance customer retention and loyalty.

Recommender systems will be helpful in most enterprises, and that we will encourage everybody to be told a lot of regarding this intriguing field.

6. References

1 N. Nassar, A. Jafar, Y. Rahhal. A novel deep multi-criteria collaborative filtering model for recommendation system.Knowledge-Based Systems, 187 (2020), Article 104811

2 I Ha, K J Oh, M D Hong, *et al.* Social Filtering Using Social Relationship for Movie Recommendation[C]// International Conference on Computational Collective Intelligence: Technologies & Applications., Springer-Verlag (2020)

3 O. Celma and P. Herrera, "A new approach to evaluating novel recommendations", RecSys '08: Proceedings of the 2018 ACM conference on Recommender systems, New York, NY, USA, 2018, ACM.

4 R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships atmultiple scales to improve accuracy of large recommender systems" KDD '17: Proceedings of the 13th ACM SIGKDD international conference on Knowledgediscovery and data mining, New York, NY, USA, 2017, ACM.

5 S. Sharma, V. Rana, M. Malhotra Automatic recommendation system based on hybrid filtering algorithm Education and Information Technologies (2021).

6 G. Linden, B. Smith, and J. York, "Amazon recommendations:Itemto-item collaborative filtering," *IEEE Internet Comput.*, Feb. 2018.

7 Liu H., Xia F., Chen Z., Asabere N. Y., Ma J., and Huang R., "TruCom: Exploiting domain-specific trust networks for multicategory item recommendation," *IEEE Systems Journal*, vol. 11, pp. 295–304, 2017.

8 Research and Development of Movie Social System, Qiu BinLiuShengGaoBi FengGuo (2020)

9 Xia F., Wang W., Bekele T. M., and Liu H., "Big Scholarly Data: A Survey," *IEEE Transactions on Big Data*, vol. 3, pp. 18–35, 2017.

ANNEXURE-1

Annexure-2