Project Requirement and Specification

on

Movie Recommendation System

(B.Tech VI Semester Mini project )

2021-2022



Submitted to:

Mr.Ankit Gupta ,

Ms.Vishu Tyagi Submitted by:

Aradhya Garg

Roll. No.: 2014586

Guided by: B.TECH -VI-Sem

Dr Surender Singh Samant Session: 2021-2022

(Resource Person)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA UNVERSITY, DEHRADUN

--------------------------CONTENTS-----------------------

1.1 ABOUT PROJECT--------------------------------------

1.2 SYSTEM REQUIREMENT--------------------------- 1.3 Database Analyzing, design and implementation -------------------------------------------

1.4 MODULES AND LIBRARIES USED----------------1.5 REFERENCE----------------------------------------------

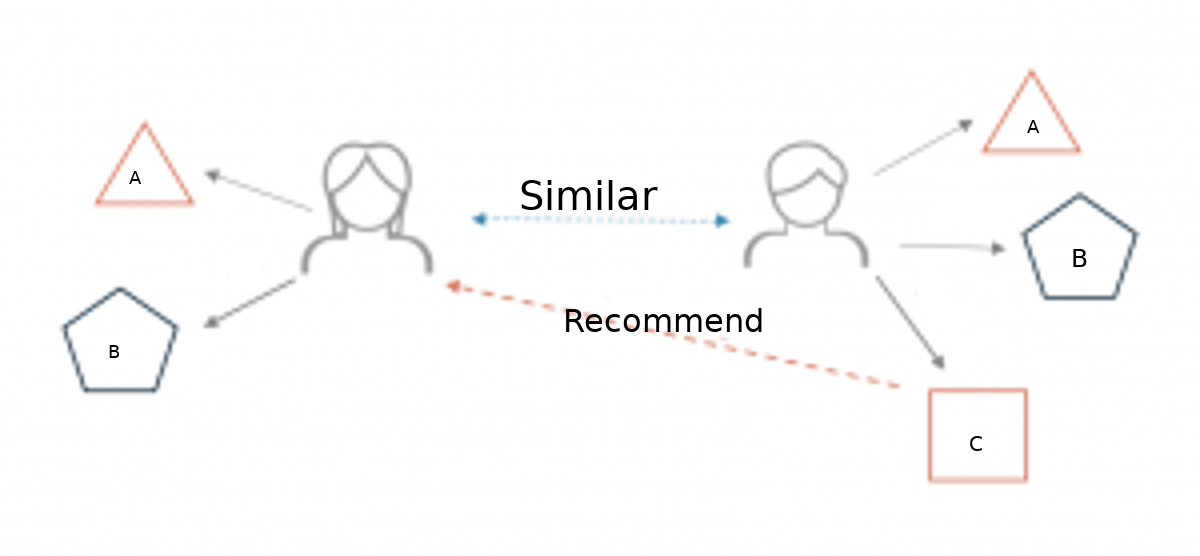
----------------------------------------------------------------------------------------

1.1 About Project

What is a recommender system?

A recommender system is a simple algorithm whose aim is to provide the most relevant information to a user by discovering patterns in a dataset. The algorithm rates the items and shows the user the items that they would rate highly. An example of recommendation in action is when you visit Amazon and you notice that some items are being recommended to you or when Netflix recommends certain movies to you. They are also used by Music streaming applications such as Spotify and Deezer to recommend music that you might like.

Below is a very simple illustration of how recommender systems work in the context of an e-commerce site.



Two users buy the same items A and B from an e-commerce store. When this happens the similarity index of these two users is computed. Depending on the score the system can recommend item C to the other user because it detects that those two users are similar in terms of the items they purchase.

Different types of recommendation engines

The most common types of recommendation systems are content-based and collaborative filtering recommender systems. In collaborative filtering, the behavior of a group of users is used to make recommendations to other users. The recommendation is based on the preference of other users. A simple example would be recommending a movie to a user based on the fact that their friend liked the movie. There are two types of collaborative models Memory-based methods and Model-based methods. The advantage of memory-based techniques is that they are simple to implement and the resulting recommendations are often easy to explain. They are divided into two:

* User-based collaborative filtering: In this model, products are recommended to a user based on the fact that the products have been liked by users similar to the user. For example, if Derrick and Dennis like the same movies and a new movie come out that Derick like, then we can recommend that movie to Dennis because Derrick and Dennis seem to like the same movies.
* Item-based collaborative filtering: These systems identify similar items based on users’ previous ratings. For example, if users A, B, and C gave a 5-star rating to books X and Y then when a user D buys book Y they also get a recommendation to purchase book X because the system identifies book X and Y as similar based on the ratings of users A, B, and C.

Model-based methods are based on Matrix Factorization and are better at dealing with sparsity. They are developed using data mining, machine learning algorithms to predict users’ rating of unrated items. In this approach techniques such as dimensionality reduction are used to improve accuracy. Examples of such model-based methods include Decision trees, Rule-based Model, Bayesian Model, and latent factor models.

Content -based systems use metadata such as genre, producer, actor, musician to recommend items say movies or music. Such a recommendation would be for instance recommending Infinity War that featured Vin Diesel because someone watched and liked The Fate of the Furious. Similarly, you can get music recommendations from certain artists because you liked their music. Content-based systems are based on the idea that if you liked a certain item you are most likely to like something that is similar to it.

Diagram

Description automatically generated

1.2 Software and Hardware Requirement Specification

1.2.1 Methods

• Collecting the data sets

• Data Analysis

• Algorithms

• Training and testing the models

• Model Deployment

1.2.2 Programming/Working Environment

• Python

• Jupyter Notebook

• MS Excel

• Front-end Development

1.2.3 Requirements to run the application

• Good internet connection

• A personal computer

• Any compiler to run python code

• Text Editor

• Chrome Driver

• Python Libraries

1.3 Database Analyzing, design and implementation

1.3.1 Key Concept

To understand movie recommendations in a little depth we need to understand the two broad types of recommendation systems which are: Content Based and Collaborative based recommendation systems.

1. Content Based Recommendation System: It uses attributes such as genre, director, description, actors, etc. for movies, to make suggestions for the users. The intuition behind this sort of recommendation system is that if a user liked a particular movie or show, he/she might like a movie or a show similar to it.
2. Collaborative Based Recommendation System: It matches users with same interests and gives recommendations based on their likes. Collaborative filtering does not make use of metadata to give recommendations.

1.3.2 System Architecture of the Proposed System

Diagram

Description automatically generated

Architecture For content based approach

1.3.2 Dataset

The total raw datasets used in the project are 2. The datasets have been collected from https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata?select=tmdb\_5000\_movies.csv.

Graphical user interface, application

Description automatically generated

1.4 Modules And Libraries Used

1.4.1 Pandas:

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

It can be installed by typing the following command in the terminal

pip install pandas

1.4.2 Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It can be installed by typing the following command in the terminal

pip install numpy

1.4.3 Pickle

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream.

It can be installed by typing the following command in the terminal

pip install pickle-mixin

pip install pickle-mixin

1.4.4 Streamlit

Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc.

1.4.5 Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

1.4.6 Nltk

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

# 1.5 REFERENCE

1. GEEKS FOR GEEKS.
2. STACK OVERFLOW.
3. BOOKS : Python Crash Course