*A Mini Project report on*

MOVIE RECOMMENDER USING PYTHON

*Submitted in partial fulfillment of the course*

CSE-1006: Foundation of Data Analytics

Under Guidance of Prof. Deepasikha Mishra

By

19BCE7044 CHOUTA MAHANTH

19BCD7095 JALADI HIMA VENKATA SAI SAKETH RAM



SCHOOL OFCOMPUTER SCIENCE AND ENGINEERING

VIT-AP UNIVERSITY, INAVOLU, AMARAVATI

JUNE, 2021

***INDEX***

1. INTRODUCTION

2. AIM

3. WORKING WITH DATA SET

4. EXTRACTING DATA

5. DATA CLEANING

6. DATA SORTING

7. PREDICTION/ANALYSIS USING ML TECHNIQUE

8. RESULTS

9. PLOTS

10. CONCLUSION



**1) INTRODUCTION:**

Recommendation systems have become quite popular nowadays. Recommendation systems have been around with us for a while now, and they are so powerful. They do have a strong influence on our decisions these days. From movie streaming services to online shopping stores, they are almost everywhere we look. Some of these are present even in our video streaming apps, shopping apps etc. These help us to go from one product to another or one movie to other based on our search. These are based on different types of recommendation systems.

There are three types of recommendation systems — Content-based, Popularity based and Collaborative based. Popularity based is simple and recommends based on the popularity of the product of video or movie or product. Content-based systems are based on the content of the movie or show and recommend similar shows. The collaborative is based on user patterns. If two users are similar and one watches a movie, the other is recommended to watch it too!

This project is one such kind of content-based recommendation of movies using a powerful tool called Python. Nice examples of this recommender are Netflix, YouTube, Disney+ and more. For example, Netflix recommends similar shows that you watched before and liked more. With this project, you will have a better understanding of how these online streaming services’ algorithms work. This will be a simple project where we will be able to see how machine learning can be used in our daily life.

**2) AIM:**

We will build a Movie Recommendation System which will take as input a movie title from the user and return the top 50 recommended movies to the user based on the input movie. This is the basic aim of this recommender.

We will use a movie dataset for our purpose. We will use the required library functions here to find the most common movie title to the one provided by the user to account for spelling mistakes and typos. This helps us to avoid errors and make our model more usable.

We find out similarities so as to return to the user the top 50 movies they should watch after the one they provided. It is a content-based recommendation system and works pretty well!

**3) WORKING WITH DATA SET:**

Data set in data science-related projects are very important because everything depends on them. The entire program working depends on how we construct the data set. Our model will be trained multiple times if a dataset is involving images or videos to improve accuracy. We are working with the data set which was acquired by the responses of nearly 5000 volunteers. As our data set contains only responses but no images or videos there is no need to execute it those many times. Executing once, the accuracy of our project is improved very well. Our working with the data set involves importing the whole data after that extracting the columns which are required by our project. The next step is to clean it and proceed with our project.

To study the heavy data sets in data science projects we have some particular library functions to do that. In this project the libraries we use are pandas, numpy, CountVectorizer, cosine\_similarity, difflab.

**pandas -** pandas is a software library written for the Python programming language for data manipulation and analysis.

**numpy-** numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

**CountVectorizer-** CountVectorizer is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text.

**cosine\_similarity-**  cosine\_similarity (X, Y=None, dense\_output=True)[source] Compute cosine similarity between samples in X and Y. Cosine similarity, or the cosine kernel, computes similarity as the normalized dot product of X and Y: K(X, Y) = <X, Y> / (||X||\*||Y||) On L2-normalized data, this function is equivalent to linear\_kernel.

**difflab-** difflib is a Python module that contains several easy-to-use functions and classes that allow users to compare sets of data. The module presents the results of these sequence comparisons in a human-readable format, utilizing deltas to display the differences more cleanly.

The first thing we have to do for our project is to import them into our kernel.

We have to import pandas into some variable as it makes our work easy which is pd and numpy is imported into kernel via np variable

CountVectorizer and cosine\_similarity are present scikit learn library and they are imported into kernel and the last one difflib is also imported.

**4) EXTRATCING DATA:**

Extracting data means reading the whole data set we collected into our program using some inbuilt library function. We will use pandas read\_csv for doing it.

After reading let us confirm our kernel have done it or not. So let us see what our kernel have read the data set. For that, we use the head() function which prints the first 5 rows of our dataset.

**5) DATA CLEANING:**

From the above step we have seen that the data set which we have used consisted of 24 columns. All the columns in the dataset are not necessary. We have to use only some specific columns to our “MOVIE RECOMMENDER”.

In the first part of cleaning the data we will remove the duplicates in the title columns so that all the duplicate movie names will be removed. We will use a simple function called drop\_duplicates() .

Let us see whether the movie name duplicates were removed or not.

We will see that top 20 movie names were printed with a number order.

Then we can see its headers and look up the data to determine which features are useful in recommendations and which are not. We then filter out headers which we are going to use in our classifier. The next step involves of selecting the columns we want. We will select four columns namely keywords, cast, director, tagline.

There may be empty spaces in our selected columns. We will fill out all empty rows in those columns with an empty string. It replaces null (NaN) values with 0.

Then we need to merge data from all those columns into one. So, we define a function to combine those features into one and name that column as combined\_features. We will be using this column for building our model. Then we will append this column to our dataset.

Now if you print the header part of our data set we will be able to see the combined\_features column.

Now that our dataset is cleaned we can proceed to further.

**6) DATA SORTING:**

Next, we will define two helper functions which will sort out the entire data to give us the movie preferences we want.

First one is to be able to find the movie title if given its index.

For the second one (finding index from movie title), we need to be a bit cautious as sometimes users can give input movie which could have a typing error or spelling mistake or incomplete name of the movie. If the program search for the movie without sorting out or improving itself it throws error as it cannot find the wrong movie name given by user. To avoid these, we will use difflib library’s function to get the closest matches to our user’s input from our data set and then use that movie title to find its index to avoid those errors.

These pointers will move around the whole data set and return all the movie names similar to given input. As we have discussed earlier we will accept only first 50 of them.

**7) PREDICTION/ANALYSIS USING ML TECHNIQUE:**

The machine learning technique which we are using in our project is Hierarchical Clustering algorithm. Hierarchical clustering algorithms group similar objects into groups called clusters. There are two types of hierarchical clustering algorithms.

1) **Agglomerative** — Bottom up approach. Start with many small clusters and merge them together to create bigger clusters.

2) **Divisive** — Top down approach. Start with a single cluster than break it up into smaller clusters.

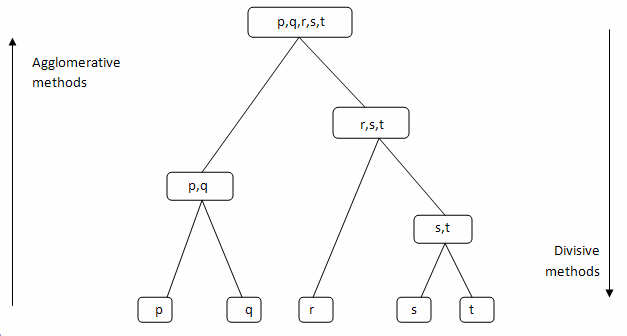


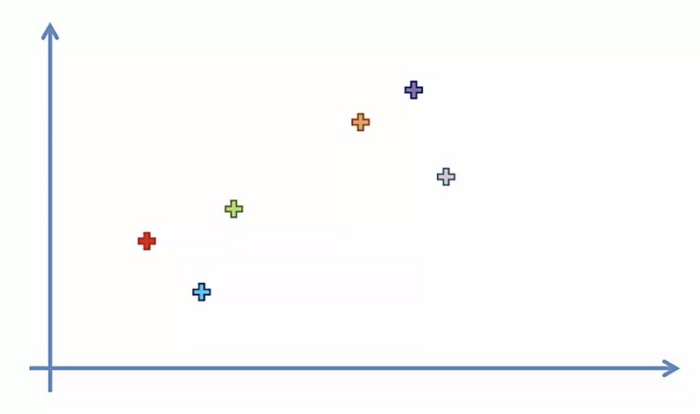
Figure 1. Depicting the both clustering algorithms

But in our project we want all the clusters collected and to be printed according to the movie input given. So we use Agglomerative Clustering algorithm as all the similar movies are collected at the last. We cannot use Divisive Clustering algorithm because it divides as it proceeds but not collects.

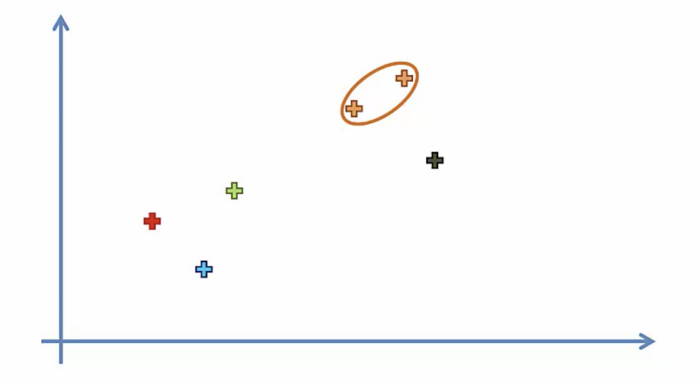
This algorithm starts at the last of our data set and proceeds to top collecting all the clusters of data it matches the input.

**How it works:**

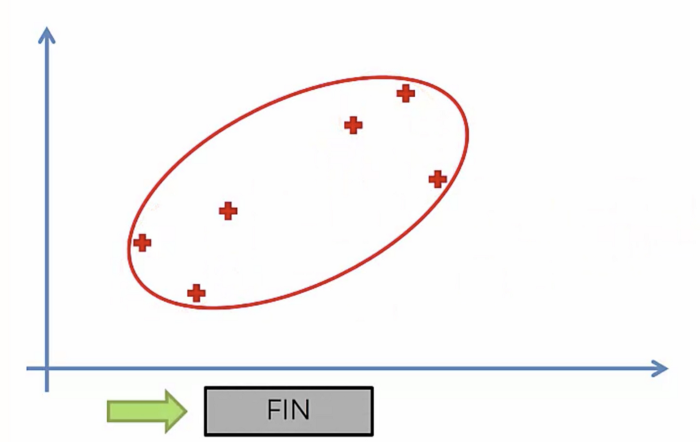
**1)** Make each data point which matches input as a cluster.



**2)** Take the two closest clusters and make them one cluster.



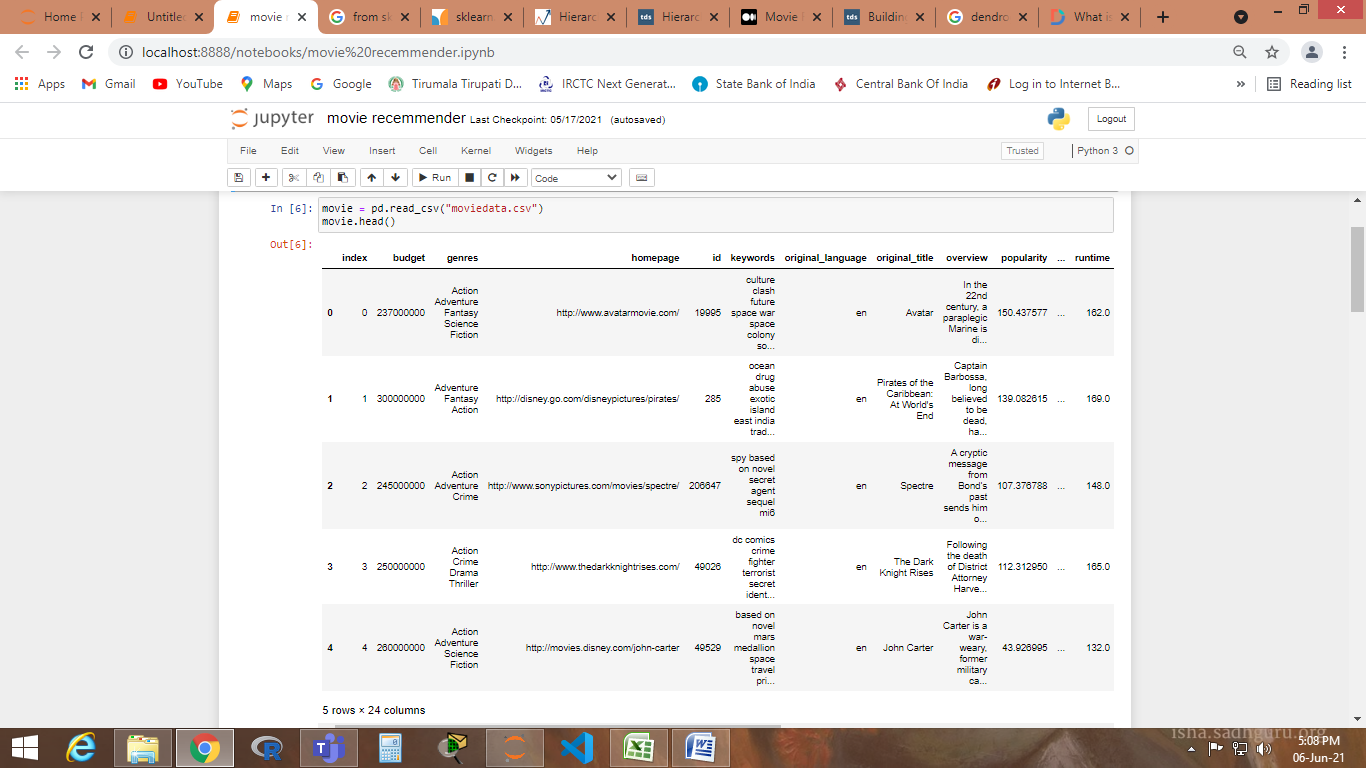
**3)** It repeats the same thing until everything similar becomes one cluster.



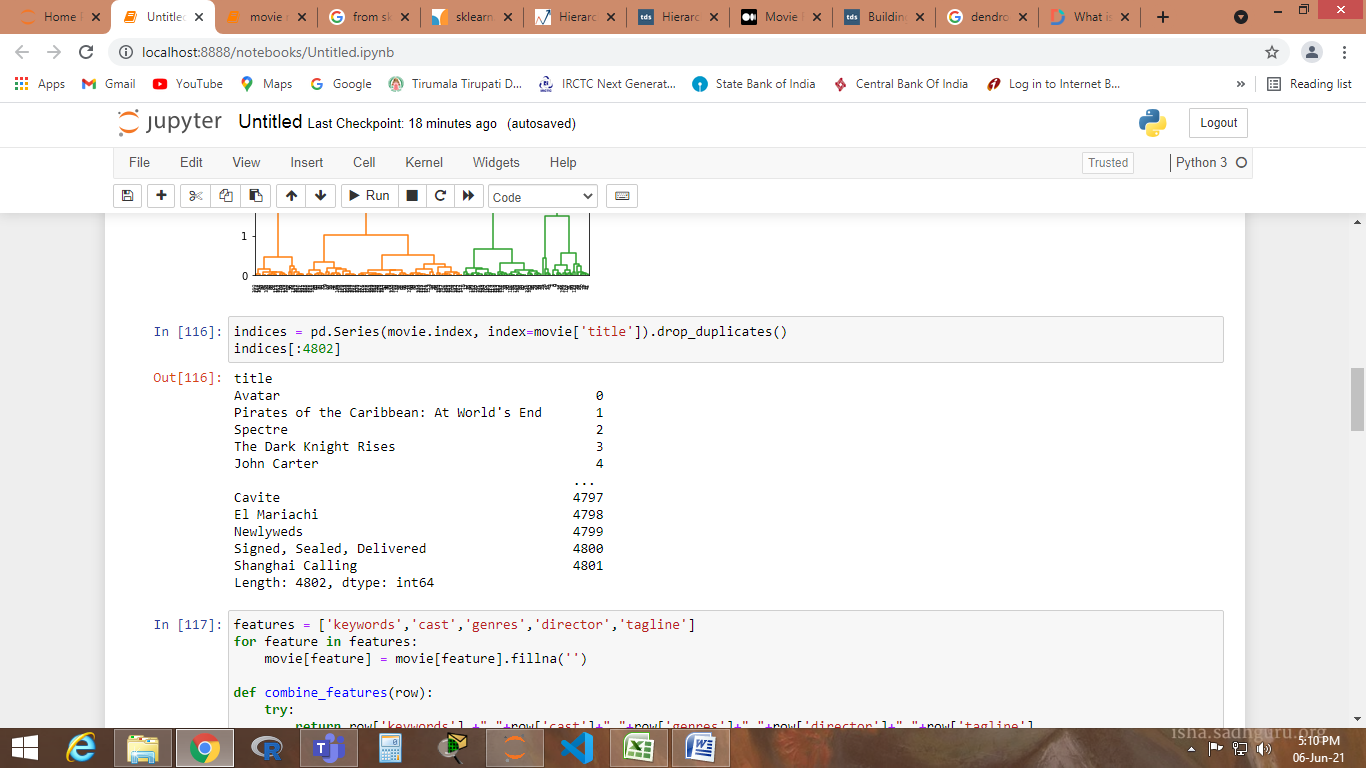
**8) RESULTS:**

Finally the outputs.

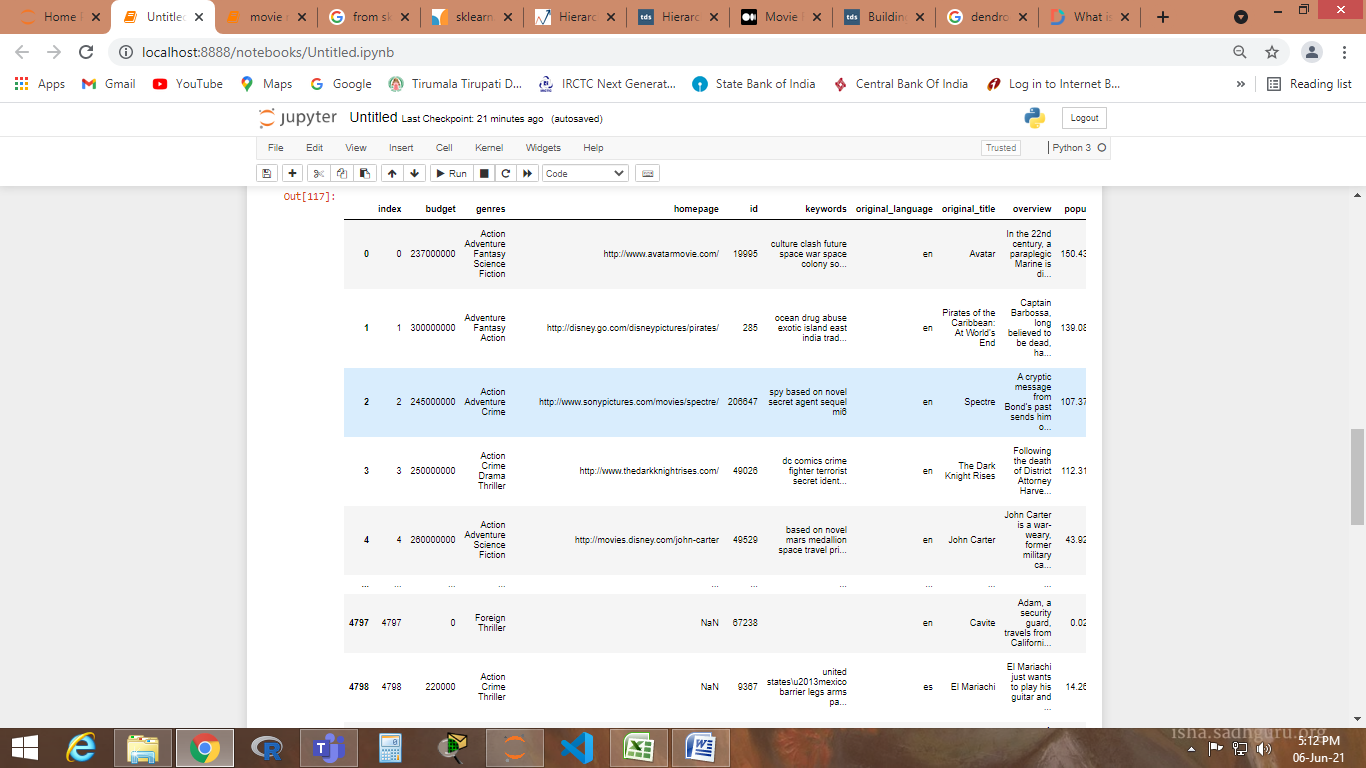
1) Printing the head of the dataset.



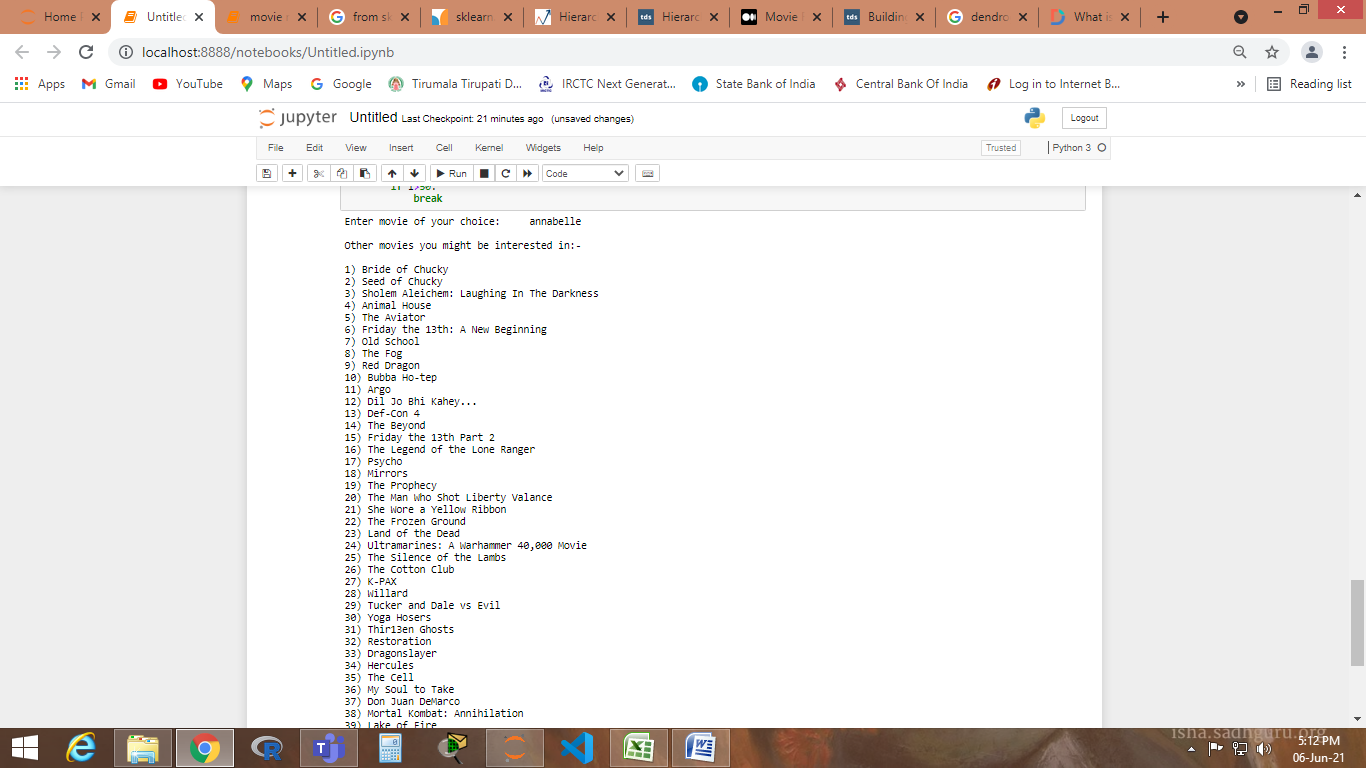
2) Giving the movies indicies.



3) Printing dataset after merging.



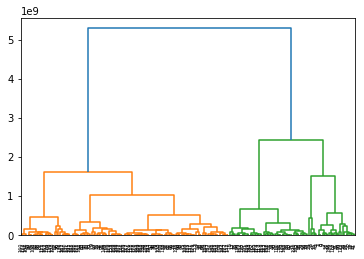
4) Finally result.



**9) PLOTS**

We plot the relation between the budget for the movie and the collection for that movie using Dendogram.

A Dendrogram is a diagram that shows the hierarchical relationship between objects. It is most commonly created as an output from hierarchical clustering. The main use of a dendrogram is to work out the best way to allocate objects to clusters.



**10) CONCLUSION:**

Recommendation Systems are the most popular type of machine learning applications that are used in all sectors. They are an improvement over the traditional classification algorithms as they can take many classes of input and provide similarity ranking based algorithms to provide the user with accurate results. These recommendation systems have evolved over time and have incorporated many advanced machine learning techniques to provide the users with the content that they want.

Through this project we can conclude that these technologies are still making the man kind very easier to access everything around him. This project also is an example of the recommender system used by many OTT apps for recommending movies based on our search.

---THE END---