**DMPA MINI PROJECT REPORT**

**Project Title: Movie Recommendation and Rating Predictor Using K Nearest Neighbors**

***Submitted by***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl#** | **Full Name** | **Reg#** | **Roll Number** | **Lab Batch** |
| **1** | **Sankeerth S** | **190953106** | **32** | **CCE 2** |
| **2** | **Shaun Santosh** | **190953232** | **56** | **CCE 2** |
| **3** | **Kura Sai Satya Varshith Reddy** | **190953266** | **60** | **CCE 2** |

***In partial fulfillment for the award of degree of***

**B. Tech**

**IN**

**COMPUTER & COMMUNICATION ENGINEERING**



**Department of Information & Communication Technology**

**December 2021**

**Abstract:**

A recommendation system attempts to predict a user's preference or rating for an item. They are becoming increasingly important as people seek convenience and are always looking for products/services that are best suited to them. As a result, recommendation systems are important because they assist them in making the best decisions. The main goal of this mini project is to provide a personalized experience for each user by showing movies based on the movie of their choice and its imdb ratings.

**Introduction:**

This recommender system uses content-based filtering by implementing the K-Nearest Neighbors algorithm. This filtering method is typically based on collecting and analyzing information about a movie’s various details such as keywords, actors, genres, and predicting what the user will like based on the movie’s similarity to other movies. The main advantage of the content-based filtering approach is that the model doesn't need any data about other users since the recommendations are specific to this user. This makes it easier to scale to many users, so it can accurately recommend complex elements such as movies. Approaches such as collaborative filter algorithms face issues such as scalability, sparsity, and cold start, which can be avoided to some extent in the proposed framework.

**Literature Survey:**

* R. Ahuja, A. Solanki and A. Nayyar, "Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor," , in this paper we mainly referred it to get an idea as to how to approach a recommendation system and figure out the different steps to the process. They have used K-nearest-neighbors in both content based as well as collaborative filtering. Based on this we decided to go with content based as it was more independent of other users, and only needed one main dataset. They have also used k-means clustering to classify the different movies, which we have avoided, as we felt it was unnecessary for our expected result.
* G Geetha1, M Safa1, C Fancy1 and D Saranya2 G “A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System”. In this paper, similar to the previous reference, it goes about implementing the problem using a hybrid approach, using both filtering methods. We have explored only the content based filtering aspect of this paper, but this research planned on further perfecting the obtained results by using collaborative filtering as well.

**Methodology**

Flow Chart:

**Diagram

Description automatically generated**

Import the python libraries:

Graphical user interface, text, application

Description automatically generated

Here the main modules we work on are pandas and scipy. Pandas is a powerful tool used for data analysis and scipy is a library used to solve scientific and mathematical problems. Json is used initially while data cleaning and operator gives us some more math functions to work with.

Import the dataset:

A picture containing text

Description automatically generated\

We import the datasets required to run the model. In this project we used ‘The movie data base’ which consists of 4000 different movie titles along with many other columns of attributes. We have decided to use 4 of attributes for the project; Genres, director and keywords from the movies file and cast from the credits file.

Cleaning the columns:

Text

Description automatically generated with medium confidence

We convert the columns in json format to strings and write all the genres from the dataset into a list.

Text

Description automatically generated

We now clean up the data we have. Above we can see that the genre column is being cleaned and sorted. Initially any leading or trailing square brackets are removed and subsequently and blank spaces or apostrophes in between the data is removed. The data is then put back into the genres column of the dataset.

Text

Description automatically generated

All unique genres in the dataset are then appended to a new list.

Graphical user interface, text, application

Description automatically generated

Here we create a new binary list, and if a genre is applicable to a certain movie a 1 is placed in the list, and a 0 if it is not applicable. We use the lambda function to go through the rows and apply the function binary to each movie. The result can be seen below.

Graphical user interface, text, application

Description automatically generated

From the binary list that was created for all the rows we can see the assignment of genres. For example, for the movie with id 0, we can see that the genres Action, Adventure, Fantasy, Science Fiction are applicable.

This process is repeated for each column until a satisfying number of attributes are achieved. In this project we clean and create 4 different binary lists similar to the one above.

Finding similarity between movies:

A picture containing text

Description automatically generated

We write a function to compute the similarity between two movies using cosine similarity. The Cosine similarity measures the cosine of the angle between two vectors projected in space. It is helpful when plotted on a multi-dimensional space, where each dimension corresponds to a genre for example in this case. The output of the above code takes the cosine distance between two movies based on genre, cast, director and keywords.

Predicting score and grouping movies:

Text

Description automatically generated

Finally, we have the function for predicting score which also finds the K nearest neighbors of the input movie. We took K = 7 for our program. After finding the nearest 7 neighbors based on cosine similarity, the average of the ratings is taken and displayed as the predicted rating of the input movie.

**Results**

We can see the output for the two movies, Avatar and Interstellar below.

Text

Description automatically generated

The result for interstellar using a front end using the Django framework

Graphical user interface, website

Description automatically generated

Text

Description automatically generated

The results of the program are very satisfactory. From these 2 results we can see that the recommendations are quite accurate, giving results related to both the common genres as well as the directors. The predicted rating is also quite accurate, with an error around .3, and it can be rectified with further tweaks in the weightage given for each aspect of the movie.

**Conclusions:**

The project explores the workings of recommendation systems and touches upon the topic of content-based filtering and K nearest neighbors. It gives us insight into how the various systems in our life make tasks much easier for us. We also can see that these systems can be surprisingly accurate without training. In conclusion, there are various categories of systems that can be implemented in different ways using varied algorithms, all to make tasks easier and life better.

**References:**

* G Geetha1, M Safa1, C Fancy1 and D Saranya2 G *“A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System”* Geetha et al 2018 J. Phys.: Conf. Ser. 1000 012101
* R. Ahuja, A. Solanki and A. Nayyar, "Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor," *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2019, pp. 263-268, doi: 10.1109/CONFLUENCE.2019.8776969.
* B. Chikhaoui, M. Chiazzaro and S. Wang, "An Improved Hybrid Recommender System By Combining Predictions, " IEEE Workshops of International Conference on Advanced Information Networking and Applications, pp. 644-649, 2011.