**Recommended Movie Prediction System – Synopsis**

**Abstract**

Movie Recommendation system is a filtration program to predict the rating or preference of a user towards a domain specific movie or genre. So, when the user provides some data about self, the data is processed against a wide variety of Movie Sets where the ratings, genre details are already pre-recorded. For which, we use the largest Movie Dataset from MovieLens to do the prediction.

Almost everyone today uses technology to stream movies and television shows. While figuring out what to stream next can be daunting, recommendations are often made based on a viewer’s history and preferences. This is done through machine learning and AI predictions. This is implemented in Python with data from the MovieLens Dataset. Generated by more than 6,000 users, MovieLens currently includes more than 1 million movie ratings of 3,900 films.

**Advantages**

Mostly the Recommendation system will be Movie Predictor for the user as per the past preferences of the user. User will be able to select the movies of their preferences, choices and according to that, the app dashboard will prefer more such kind of movies for the user.

But here, we will use collaborative Filtering technique where we will compare the user preferences with other users’ behaviour which are already recorded in the data set and is evolving as per their suggestions and usage of the app.

**Project Scope**

The project scales an ML/AI System for predicting Movies for users according to their Personalized Preferences and Ratings. Also, the system will provide the facility to create and read reviews and ratings. The MovieLens Dataset will be updated with recent Movie Details constantly with the help of ML. The scope of the project extends to the usage of Users and the Movie Updates on a daily basis.

**Modules**

1. Authentication
2. Movie Details
3. Movie Prediction
4. Review Management
5. Personalization
6. Movie Schedule List

**Module Description**

1. **Authentication**

We will use Firebase Authentication to enable access to the System. Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

1. **Movie Details**

In this module, User and App Admin can Add/remove Movies, Movie details which will be used for Prediction Algorithms. The admin can add new movies to the portal through his exclusive interface. The module includes each movie to the database whenever a MovieLens adds a new movie to the dataset under supervision of the admin.

1. **Movie Prediction**

Movie Prediction system utilises MovieLens dataset to interpret and improvise the prediction using personalised algorithms which will provide most accurate predictions on which a user wants to or need to see next. According to that the system will suggest movies to the user, reviews and rating to user. The ML model will adapt itself as if we add more movies, reviews and ratings. We will use Collaborative Filtering technology to implement the prediction system.

**User-based Collaborative filtering**

The basic idea here is to find users that have similar past preference patterns as the user ‘A’ has had and then recommending him or her items liked by those similar users which ‘A’ has not encountered yet. This is achieved by making a matrix of items each user has rated/viewed/liked/clicked depending upon the task at hand, and then computing the similarity score between the users and finally recommending items that the concerned user isn’t aware of but users similar to him/her are and liked it.

For example, if the user ‘A’ likes ‘Batman Begins’, ‘Justice League’ and ‘The Avengers’ while the user ‘B’ likes ‘Batman Begins’, ‘Justice League’ and ‘Thor’ then they have similar interests because we know that these movies belong to the super-hero genre. So, there is a high probability that the user ‘A’ would like ‘Thor’ and the user ‘B’ would like The Avengers’.

**Item-based Collaborative Filtering**

The concept in this case is to find similar movies instead of similar users and then recommending similar movies to that ‘A’ has had in his/her past preferences. This is executed by finding every pair of items that were rated/viewed/liked/clicked by the same user, then measuring the similarity of those rated/viewed/liked/clicked across all user who rated/viewed/liked/clicked both, and finally recommending them based on similarity scores.

Here, for example, we take 2 movies ‘A’ and ‘B’ and check their ratings by all users who have rated both the movies and based on the similarity of these ratings, and based on this rating similarity by users who have rated both we find similar movies. So if most common users have rated ‘A’ and ‘B’ both similarly and it is highly probable that ‘A’ and ‘B’ are similar, therefore if someone has watched and liked ‘A’ they should be recommended ‘B’ and vice versa.

1. **Review Management**

Users will be able to add/remove Reviews and ratings as per the movie they have watched. And it will provide a forum to discuss about the same.

1. **Personalization**

User Personalization system allows what kind of movies they are expecting, the movies they like to watch, genre, and other perspectives of a movie.

1. **Movie Schedule List**

This Module helps Users to schedule movies to watch from the list, and it will remind users with prompt notifications on time to make sure they don’t miss their favourite bucket list.

**Technologies used:**

Frontend: React.Js, Node.js, OAuth2.0 Authentication

Backend and ML Model Training: Python (Django Web), MongoDB

ML Dataset: MovieLens

ML Model: Customised