

What are Recommender system

A recommender system is a type of software application or algorithm that provides personalized suggestions or recommendations to users. The primary goal of recommender systems is to predict the preferences or interests of users and recommend items (such as products, movies, music, articles, or other content) that are likely to be of interest to them. These systems are widely used in various online platforms to enhance user experience and engagement.

There are several types of recommender systems, and they can be broadly categorized into the following:

Collaborative Filtering:

- User-Based Collaborative Filtering: This approach recommends items based on the preferences of users who are similar to the target user. If two users have similar preferences on certain items, the system may recommend items liked by one user to the other.
- Item-Based Collaborative Filtering: This approach recommends items that are similar to those liked by the target user. It relies on item-item similarity metrics.

Content-Based Filtering:

- This approach recommends items based on the characteristics and features of the items, as well as the preferences expressed by the user. It involves creating user profiles and item profiles and recommending items that match the user's preferences.

Hybrid Recommender Systems:

- These systems combine multiple recommendation approaches to improve the accuracy and coverage of recommendations. For example, a hybrid system may integrate collaborative filtering and content-based filtering.

Here in this project, we will try to make a project based on the content based recommended system.

PROJRCT FLOW:

Step 1 : Data observation

Step 2 : Data processing

Step 3 : Model making

Step 4 : Making product from the model (website in out case)

Step 5 : Deployment

Step 1 : Data Observation

Data : <https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata>

Here from this data we will get two data set :

1. Movies
2. credits

Movies : we will majorly need all the columns from this data set

Credits : from here only "cast" and "crew" column will be required

SO WE WILL MERGE 2 DATA SET FOR THE EASY OF WORKING AND MAKE : movie_credit

Step 2: Data Processing :

Understanding the DataSet and removing the unnecessary columns from the movies_credit dataset

Now which column to remove :

1. We will just try to find which column is important from making tag for the movie.

- a. Now here making tag means to see which columns will help in making the decision while recommending. Ex:
 - i. The first column -> "Budget".
 - You can suggest someone movie based on the high budget just because his last some movies he saw was high in budget.
 - There is a chance that i might like Avatar (which is a very high budget movie) but i don't like DC movies which are again high in Budget segment.
 - b. Similarly we will try to conclude for all the columns and decide whether they are needed or not.
 - c. Here we will be ignoring numerical column just to have sort of easy work flow.
2. The new movies_credit dataset has column:
 - a. genres
 - b. id
 - c. keywords
 - d. title
 - e. overview
 - f. cast
 - g. crew
3. Now we will convert this new data set into 3 column data set , which will comprise of the following columns:
 - a. Movie_id
 - b. Title
 - c. Tag : this will be a combine column of all other column.

“””

CONCEPT:

NOW WHAT OUR AIM IS TO FIND SIMILARITY BETWEEN TWO TAGS. WE CAN BE

1. BY COMPARING AND BRING OUT THE SIMILAR WORK BETWEEN TWO TAGS [which is not good]
2. VECTORIZATION

- a. SO WHAT YOU HERE DO IS YOU PLOT YOU TAG ON THE GRAPH OR IF I SAY IT WAY AROUND I MAKE VECTOR OF THEM WHICH I PLOT ON GRAPHS
 - b. NOW AFTER THE PLOTTING WHEN YOU SELECT A MOVIE, YOU PLOT IT ON THE GRAPH AND BY THAT GRAPH YOU TRY TO GET THE SIMILAR MOVIE
 - c. NOW HOW TO GET WHICH MOVIES ARE SIMILAR ??
 - i. So the movie most closest to your select movie is the similar movie!
3. Now VECTORIZATION has many technique:
- a. Bag of word [using this]
 - b. Tfidf
 - c. word2vec
4. BAG OF WORDS:
- a. Here you concatenate all the tags making a mega tag.
 - b. Then we will hash map the frequency of all the words which occurred in that mega tags as the key [here we will not keep stop word for this hash map keys].
 - c. Now you will again take all the tag you made for each movie and from that tag you will try to fill this hash map now every movie's tag will form a list of number of how many time a word has occurred in that tags This list of number is the vector fot that tag.