Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Bachelor of Science in Computer Science and Engineering

Course No: CSE 4108

Course Title: Artificial Intelligence Lab

Report: Movie Recommendation System.

Date of Submission: 04/09/2022

Submitted to:

Ms. Tamanna Tabassum
Lecturer, Department of CSE, AUST.

Mr. Md. Siam Ansary

Lecturer, Department of CSE, AUST.

Submitted by,

Name: Sanjay Kumar Mandal

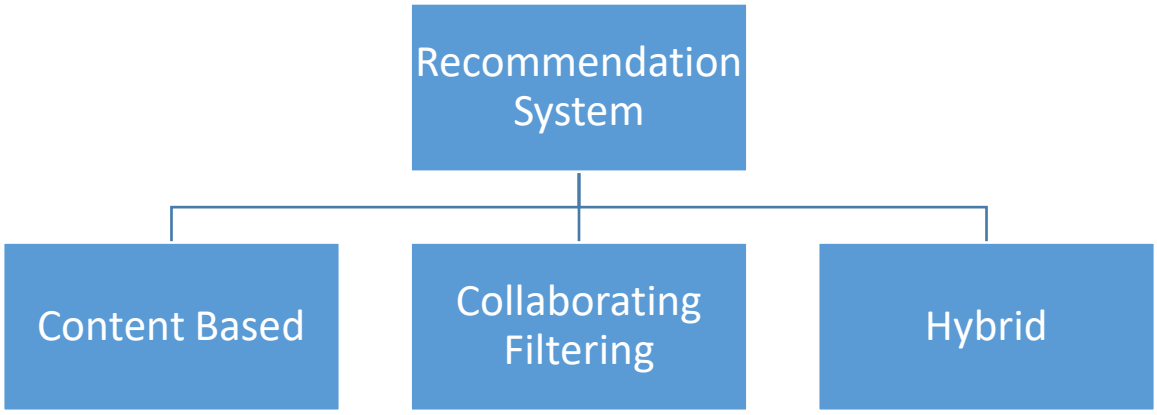
Student ID: 18.02.04.039

Movie Recommendation System

Report

Description:

A Recommendation System is a type of system where depending on the input given by the user, he/she can see similar type of content. There are three types of recommendation systems. They are given below:



In my project movie recommendation system I have implemented content based recommendation system. Here the contents are the tags that is generated from specific attributes. Depending upon the input this system will show 10 similar movies as output.

Dataset description:

There are 2 dataset used. One for movies and another one is for credits. These two datasets are from tmdb which is a database for movies. These datasets contains 5000 samples. Movies dataset contains 20 attributes and credit database contains 4

attributes. Both datasets has two attributes common which are movie_id and title. The attributes are described below:

- **Movie database:**

Size:

5000 samples.

Attributes:

- Budget: Budget of movies.
- genres: Genre of movies.
- homepage: Webpage of movies.
- id: Unique id of movies.
- keywords: Word helps to search.
- original_language: Original language of movies.
- original_title: Original title of movies.
- overview: Short overview of movies.
- popularity: Popularity among fans.
- production_companies: Production companies involved.
- production_countries: Countries where filmed.
- release_date: Release date of movies.
- revenue: Revenue of of movies.
- runtime: Runtime of of movies.
- spoken_languages: Languages that are spoken in movies.
- status: released or unreleased.
- tagline: One sentence that defines the movie.
- title: Title of movies.
- vote_average: Vote of movies.
- vote_count: Vote count of movies.

- **Credits database:**

Size:

5000 samples.

Attributes:

- movie_id: Unique id of movies.
- Title: Title of movies.

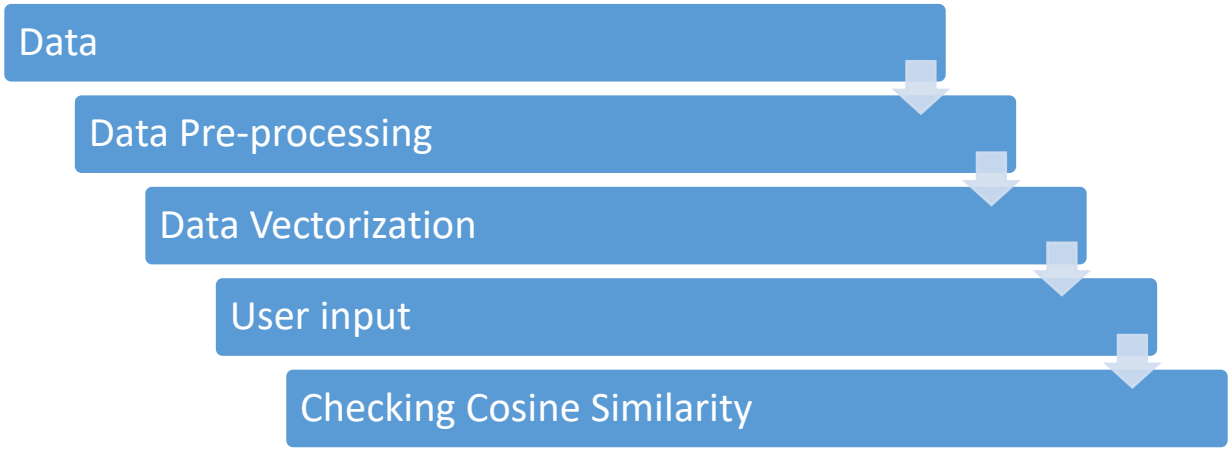
- Cast: Cast of movies.
- Crew: Crew of movies.

Reference:

https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata?select=tmdb_5000_movies.csv

Working procedure:

The work flow diagram of this project is given below:



Here first 2 database files are imported. Then we pre-process the data by merging two database into one on basis of 'title' attribute. Then process data by eliminating redundant rows which are not required for recommendation systems. We only choose movie_id, title, overview, genres, keywords, cast and crew attributes which are required attributes for the system. Then we convert string values into list using convert() function. We also get original cast name from convert_original() function.
Likewise, director is fectched and we remove all the spaces. Following that concatenation is done. After that some preprocessing is also done like lower casing all characters, stemming every word using stem() function.

We can’t use regression as the tags column is in text. So we use vectorization in tags. Every movie will become a vector. If user picks one movie then system will recommend closest vectors. Popular text vectorization technique is **Bag of words**.

Models:

❖ Bag of words:

In my project I used bag of words model. This is a technique used in natural language processing. First we concate tags thus a large string will be returned. In that string this model will search 5000 common word and frequency will also be counted of those words.

These 5000 common words will be compared to all our movies respectively and count how many times those common words are repeated. Let’s say, ‘action’ is a word which is repeated in a particular movie for 5 times. Likewise, this procedure will be done for every movie and every word and storing these frequency values for vectorization purpose.

Finally cosine similarity is used to calculate the closest value of the given input movie and the closest 10 movies will displayed accordingly.

Discussion:

In this project I implemented one model which is Bag of words which is pretty accurate. As I have only used one model there is no comparison. From the output as a movie fan myself it gives satisfactory result from the given dataset.