Project Report

Name: Adhya Borisa

Batch: DST 20823

Project name: Movie Recommendation System

Introduction:

What is recommendation system?

A recommendation system provides suggestions to the users through a filtering process that is based on user preferences and browsing history.

The information about the user is taken as an input. The information is taken from the input that is in the form of browsing data. This information reflects the prior usage of the product as well as the assigned ratings.

A recommendation system is a platform that provides its users with various contents based on their preferences and likings. A recommendation system takes the information about the user as an input. It is a simple algorithm whose aim is to provide the most relevant information to a user by discovering patterns in dataset.

Types of recommendation systems:

1) Content-Based Movie Recommendation Systems

Content-based methods are based on the similarity of movie attributes. Using this type of recommender system, if a user watches one movie, similar movies are recommended. For example, if a user watches a comedy movie starring Adam Sandler, the system will recommend them movies in the same genre or starring the same actor, or both. With this in mind, the input for building a content-based recommender system is movie attributes.

2) Collaborative Filtering Movie Recommendation Systems

With collaborative filtering, the system is based on past interactions between users and movies. With this in mind, the input for a collaborative filtering system is made up of past data of user interactions with the movies they watch.

For example, if user A watches M1, M2, and M3, and user B watches M1, M3, M4, we recommend M1 and M3 to a similar user C. You can see how this looks in the figure below for clearer reference.

Here we are using content-based recommendation system.

Building Movie Recommendation System:

Importing libraries:

- Numpy: It provides support for arrays , matrices and mathematical functions to operate on these arrays
- Pandas: It is for data structures like DataFrames and tools for reading and processing data.
- Matplotlib.pyplot: Plotting library to visualize data.

- Seaborn: It provides a high level interface for drawing attractive and informative statistical graphics.
- Warnings: Using to handle warnings in code execution.

Next, we are loading the dataset as dataframe using pandas.

Then ,we are knowing about data by '.columns' , '.info', '.describe()' , etc.

Columns of dataset are:

```
id                    int64
title                 object
vote_average          float64
vote_count            int64
status                object
release_date          object
revenue               int64
runtime               int64
adult                 bool
backdrop_path         object
budget                int64
homepage              object
imdb_id               object
original_language     object
original_title        object
overview              object
popularity            float64
poster_path           object
tagline               object
genres                object
production_companies  object
production_countries  object
spoken_languages      object
```

Now, we are cleaning the data and removing the columns which is not necessary for analysis.

I have replaced the missing “popularity” column values with the mean of existing data.

Then, we are removing the duplicate entries from the dataset.

After cleaning the data, we are performing visualization on the data for understanding the data more better.

We are selecting the essential columns only for further process.

Selected columns are:

‘id’, ’title’, ’genres’, ’adult’, ’overview’, ’popularity’.

Not selecting remaining columns because it is not related to our model prediction.

After that we are converting the string columns ‘genres’ and ‘overview’ in list.

Some genre names contain spaces (for eg ‘science fiction’, ‘Romantic comedy’ etc.) which, if left unchanged, could affect the accuracy of recommender system. By removing spaces, we create clear consistent representation ensuring that similar genres are treated as identical entities.

We have converted the Boolean ‘adult’ column to categories because this will enable the recommender system to give recommendations based on the audience type.

Then we combined transformed data for training:

```
df['tags'] =  df['overview'] + df['genres'] + df['adult']
```

After that we are creating new dataframe ‘new_df’ for training.

```
new_df = df[['id','title','tags']]
```

Then merging the tags into single string by that the model will understand easily to recommend movie.

We are importing NLTK (Natural Language Toolkit) for using porter stemmer. Stemming is the process in natural language processing that aims to reduce words to their base form, usually by removing affixes. For example the word is ‘singing’ , ‘sings’ so it will be ‘sing’. The porter stemmer is a popular stemming algorithm that allows for the transformation of words into their base form.

Then we have provided a function named ‘stem’ to perform stemming operations on text and applied it to ‘tags’ column and also converted tags to lowercase because it ensures uniformity in the column, making the text consistent in case facilitates easier comparisons, analysis within movie recommender system.

Count vectorizer :

It is used to convert the text-based data into numerical representations to build models , helps in representing data in a format suitable for ML models

This technique is a part of text processing steps in natural language processing.

Then, we set the parameters ‘max_features’ that sets the maximum number of unique words to be extract from the text and ‘stop_words’ that declares that English stops words such as common and less informative words like ‘the’, ‘and’, ‘is’, etc., will be removed during the text transformation process.

Cosine similarity:

This is used to calculate the similarity between different movies based on their feature representations, assisting in

recommending similar movies to users based on their preferences.

At last, we are creating a ‘recommend’ function which performs retrieving the index of input movie title and calculate the similarity scores between the input movies based on the similar matrix and sorts the movies based on their similarity scores in descending order.

And we are done!

Using ‘recommend’ function:

```
recommend('Now You See Me')
```

output:

```
Java Heat
Wind River
Marauders
The Heist of the Century
Hangman
```

These recommended movies are potentially similar to ‘Now You See Me’ based on the underlying recommendation algorithm and similarity matric utilized in the recommend function.