**A PROJECT REPORT**

**on**

**“Movie Recommendation System”**

**Submitted to**

**KIIT Deemed to be University**

**For my course Data Analytics**

**BY**

**Pummy Jha - 1906261**

**UNDER THE GUIDANCE OF**

**Prof. Manjusha Pandey**



**SCHOOL OF COMPUTER ENGINEERING**

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

**BHUBANESWAR, ODISHA - 751024**

**April 2022**

**Acknowledgements**

I am profoundly grateful to Prof. Manjusha Pandey for her expert guidance and continuous encouragement throughout to see that this project meets its target since its commencement to its completion. .....................

Pummy Jha - 1906261

**Abstract**

Aim of Project

The main goal of this machine learning project is to build a recommendation engine that recommends movies to users. This R project is designed to understand the functioning of a recommendation system. I developed an Item Based Collaborative Filter. This helped me gain experience of implementing my R, Data Science, and Machine learning skills in a real-life project.

Dataset used

I have used the MovieLens Dataset. That data I have used consists of 105339 ratings in the ratings.csv file, applied over 10329 movies in the movies.csv.

Essential Libraries

recommenderlab, ggplot2, data.table and reshape2.

Data Pre-processing

After retrieving data from the movies.csv andratings.csv datasets, I observed that the userId column, as well as the movieId column, consisted of integers. Furthermore, I needed to convert the genres present in the movie\_data dataframe into a more usable format by the users. In order to do so, I first created a one-hot encoding to create a matrix that comprises of corresponding genres for each of the films. I then created a search matrix that will allow us to perform an easy search of the films by specifying the genre present in our list.

There are movies that have several genres. For the movie recommendation system to make sense of the ratings through recommenderlab, I convert the matrix into a sparse matrix. This new matrix is of the class realRatingMatrix. I then overviewed some important parameters that provided various options for building recommendation systems for movies.

**Collaborative Filtering -** Exploring similar data

Collaborative Filtering involves suggesting movies to the users that are based on collecting preferences from many other users. For example, if a user A likes to watch action films and so does user B, then the movies that the user B will watch in the future will be recommended to A and vice-versa. Therefore, recommending movies is dependent on creating a relationship of similarity between the two users. With the help of recommenderlab, I computed similarities using various operators like cosine, pearson as well as jaccard.

**Visualisation**: Similarity in data

I visualised the similarity between the users as explained in the above section as well as the similarity shared between the films.

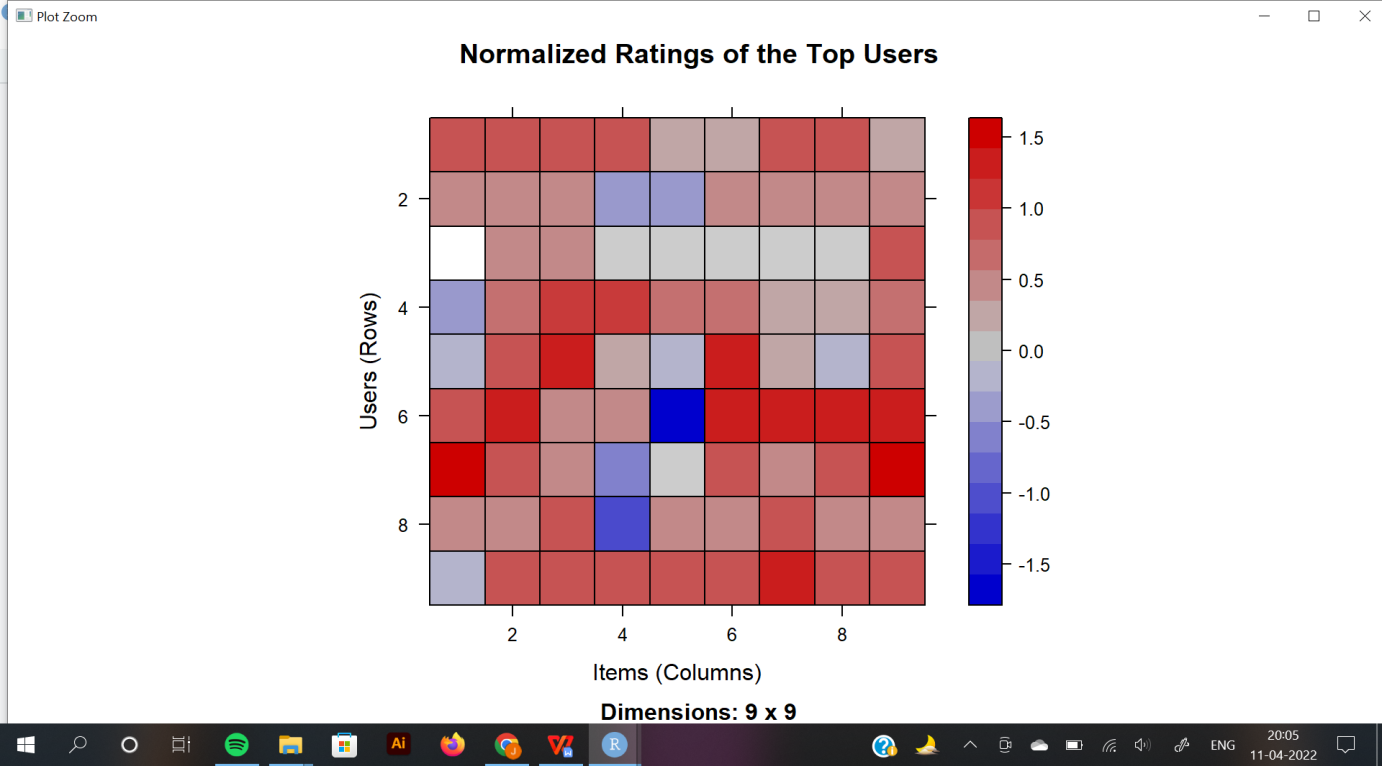
**Visualisation**: Most viewed movies

In this section of the machine learning project, I explored the most viewed movies in the dataset. Before this, I counted the number of views in a film and organized them in a table that would group them in descending order. I visualized the total number of views of the top films as a bar plot.

From the visualisation, it could be observed that 'Pulp Fiction' is the most watched film followed by 'Forrest Gump'.

**Visualisation**: Heatmap of Movie Ratings

Now, in this data science project of Recommendation system, I visualized a heatmap of the movie ratings. This heatmap will contain first 25 rows and 25 columns.



**Data Preparation**

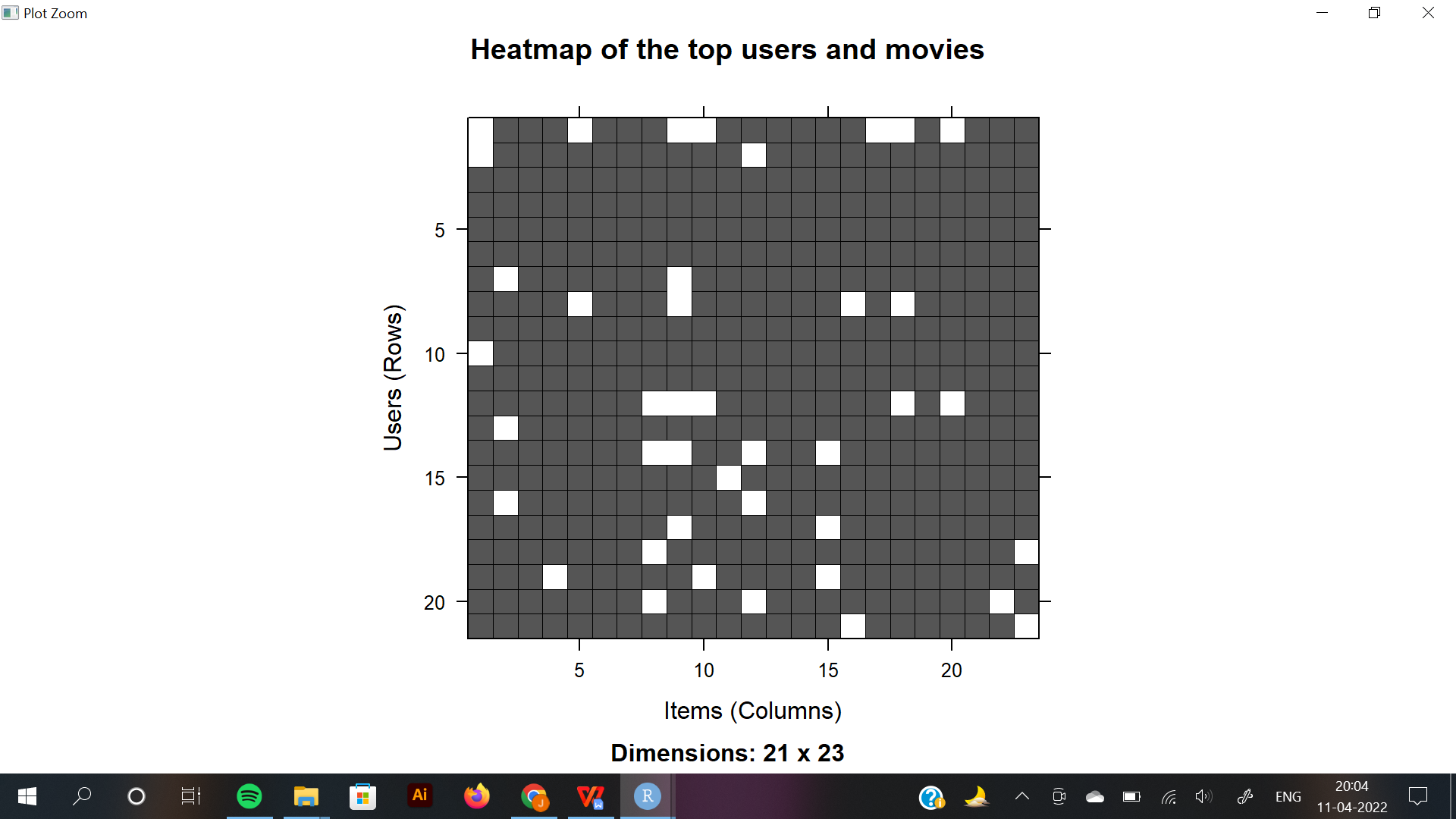
This is conducted in three steps:

Selecting useful data

Normalizing data

Binarizing the data

Data Selection: Through this I visualised the top users and movies through a heatmap. Then I visualized the distribution of the average ratings per user.



**Data Normalization**: In the case of some users, there can be high ratings or low ratings provided to all of the watched films. This will act as a bias while implementing the model. In order to remove this, I normalized the data. Normalization is a data preparation procedure to standardize the numerical values in a column to a common scale value. This is done in such a way that there is no distortion in the range of values. Normalization transforms the average value of our ratings column to 0. I then plotted a heatmap that portrays our normalized ratings.

**Data Binarization**: In the final step of the data preparation, in this data science project, I binarized the data. Binarizing the data means that we have two discrete values 1 and 0, which will allow the recommendation system to work more efficiently. I defined a matrix that will consist of 1 if the rating is above 3 and otherwise it will be 0.

**Collaborative Filtering System**

In this section of the data science project, I developed the Item Based Collaborative Filtering System. This type of collaborative filtering finds similarity in the items based on the people’s ratings of them. The algorithm first builds a similar-items table of the customers who have purchased them into a combination of similar items. This is then fed into the recommendation system.

The similarity between single products and related products can be determined with the following algorithm:

* For each Item i1 present in the product catalog, purchased by customer C.
* And, for each item i2 also purchased by the customer C.
* Create a record that the customer purchased items i1 and i2.
* Calculate the similarity between i1 and i2.

I built this filtering system by splitting the dataset into 80% training set and 20% test set.

**Building the recommendation system**

Now, I explored the various parameters of the Item Based Collaborative Filter. These parameters are default in nature. In the first step, k denotes the number

of items for computing their similarities. Here, k is equal to 30. Therefore, the algorithm will now identify the k most similar items and store their number.

**Exploring data science recommendation system model**

Using the getModel() function, I retrieved the recommen\_model. I then found

the class and dimensions of the similarity matrix, that is, contained within model\_info. Finally, I generated a heatmap, that will contain the top 20 items

and visualize the similarity shared between them.

In the next step of the ML project, I carried out the sum of rows and columns

with the similarity of the objects above 0. I visualized the sum of columns

through a distribution.

**Building Recommender System on dataset using R**

I created a top\_recommendations variable which I initialized to 10, specifying

the number of films to each user. I then used the predict() function that

identified similar items and ranked them appropriately. Here, each rating is

used as a weight. Each weight is multiplied with related similarities. Finally, I

added everything in the end.



Pummy Jha(1906261)