Movie Recommender System (Content Based)

A Mini-project Report submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications

by

Mohammad Salman

(17419MCA020)

Department of Computer Science

Institute of Science

Banaras Hindu University, Varanasi – 221005

December 2019

CANDIDATE'S DECLARATION

I **Mohammad Salman** hereby certify that the work, which is being presented in the Mini-project report, entitled **Movie Recommender System** , in partial fulfillment of the requirement for the award of the Degree of **Master of Computer Applications** and submitted to the institution is an authentic record of my/our own work carried out during the period *September, 2019* to *November-2019* under the supervision of **Dr. Gaurav Baranwal**. I also cited the reference about the text(s) /figure(s) /table(s) /equation(s) from where they have been taken.

The matter presented in this Mini-project as not been submitted elsewhere for the award of any other degree or diploma from any Institutions.

Date: Signature of the Candidate

Signature of the Supervisor

ABSTRACT

In today's world, where there is a variety of content to be consumed like books, videos, articles, movies, etc. and finding the content of one's liking has become an irksome task. This is where the recommendation system comes into the picture, where the content providers recommend users the content according to the user's liking. In this project, we use machine learning to build a movie recommendation system which will be based on the user's previous movie ratings by using the Content-Based Recommendation algorithm. Recommendation System has become ubiquitous nowadays and can be commonly seen in online stores, movies databases and job finders. This project will assist the user in identifying what kind of movie to watch out for. This project will play an important role in the content provider by allowing subscribers to watch TV shows, movies, documentaries and more on a wide range of Internet-connected devices of their liking.

TABLE OF CONTENTS

LIST OF TABLES

Table No. **Title** **Page No.**

LIST OF FIGURES

Figure No. **Title** **Page No.**

LIST OF NOTATIONS

$A < B$	A is less than B
$A > B$	A is greater than B
$A \leq B$	A is less than or equal to B
$A \geq B$	B is greater than or equal to B
E_K	Encryption key
D_K	Decryption key
F_n	*n*th Fibonacci number
I_n	the identity matrix of order n
N	Set of positive Integers
W	Set of whole numbers
Z	Set of integers
Z^+	Set of positive integers
Z_n	Set of non-negative integers less than n
μ	Static friction coefficient

CHAPTER -1

INTRODUCTION

1.1 General introduction :

In this world, peoples' tastes may vary; they generally follow patterns. By that, there are similarities in the things that people tend to like or another way to look at it, is that people tend to want something in the same category or things that share the same characteristics. For example, if you've recently purchased a book on Machine Learning in Python, and you've enjoyed reading it, likely, you'll also enjoy reading a book on Data Visualization. People also tend to have similar tastes to those of the people they're close to in their lives. Recommender systems try to capture these patterns and similar behaviours, to help predict what else you might like. A recommendation system is a type of information filtering system which attempts to predict the preferences of a user and make suggests based on these preferences. Recommender systems have many applications. Indeed, Recommender systems are usually at play on many websites. For example, suggesting books on Amazon and movies on Netflix.

Another example can be found in a daily-use mobile app, where a recommender engine is used to recommend anything from where to eat, or, what to read to. On social media, sites like Facebook or LinkedIn, regularly recommend friendships. Recommender systems are even used to personalize your experience on the web. For example, when you go to a news platform website, a recommender system will make a note of the types of stories that you clicked on and make recommendations on which types of stories you might be interested in reading. So, The popularity of recommendations systems has gradually increased.

In this project, we are going to propose a content-based recommendation system for the movie, and this movie recommendation system is correctly doing the above tasks. The main goal of this project is to give the best suggestion to the users based on their profiles

CHAPTER -2

PROPOSED APPROACH

2.1 Description of Datasets :

The Movies dataset is used in this project. It is taken from the MovieLens. The Movies data set has 9742 rows, and 3 attributes value which is named movieId, title, and genres. The description of the attribute value is given in the table below –

Table 2.2: Description of the Attributes of the Movies dataset.

S.No	Variables	Description	Type
1	movieId,	A unique id provide to every movie	int
2	title	Name of the movie	object
3	genres	Genres are identifiable types, catagories or groups of the movies	object

Table 2.2: Top 5 values of the Movies dataset:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

2.2 Recommendation System:

There is an extensive class of Web applications that involve predicting user responses to options. Such a facility is called a recommendation system. Recommendation system produces a rank list of items on which a user might be interested in the context of his current choice of an item

Recommender systems are utilized in a variety of areas and are most commonly recognized as playlist generators for video and music services like *Netflix, YouTube* and *Spotify* product recommenders for services such as *Amazon* or content recommenders for social media platforms such as *Facebook* and Twitter.

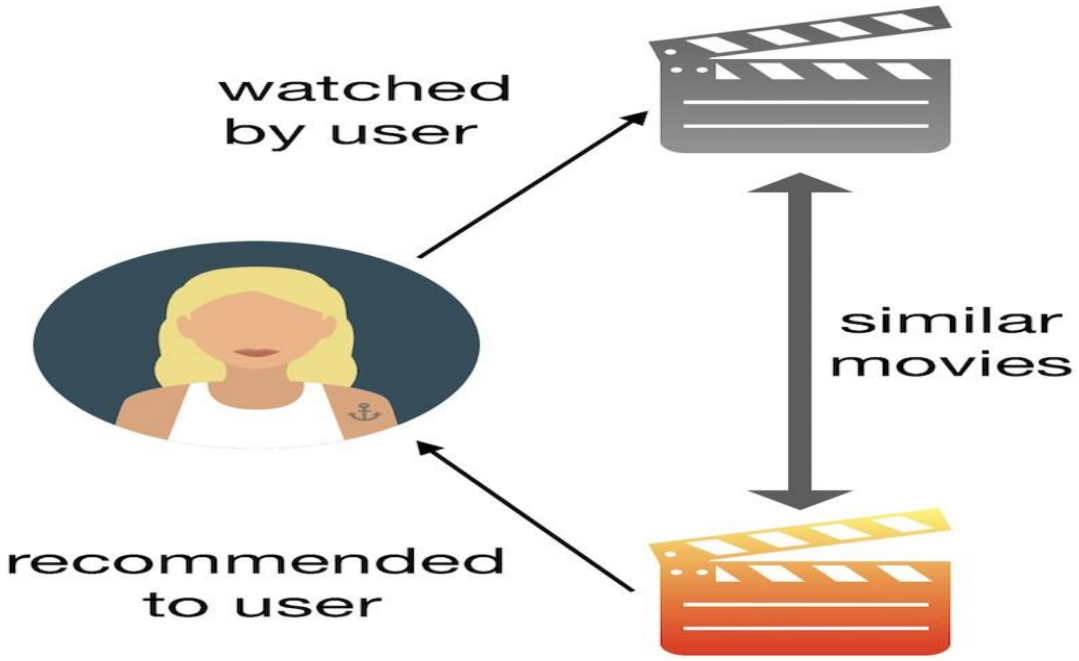
Recommendation systems use several different technologies. We can classify these systems into two broad groups.

2.2.1 Content-based recommendation systems :

Content-Based Recommendation system checks for the adores and aversions of the user and creates a User-based Profile. The user profile is the combination of the sum of the item profiles, where combination being the rating customer or user has evaluated. If a Netflix user has watched many cowboy movies, then recommend a movie classified in the database as having the "cowboy" genre.

Popular techniques in content-based filtering include the term-frequency/inverse-document- frequency (tf-idf) weighting technique in information retrieval. Advantages of Content-Based approach is that data of other users is not required and the recommender engine can recommend new items which are not rated currently.

Image 2.1- Content-based recommendation



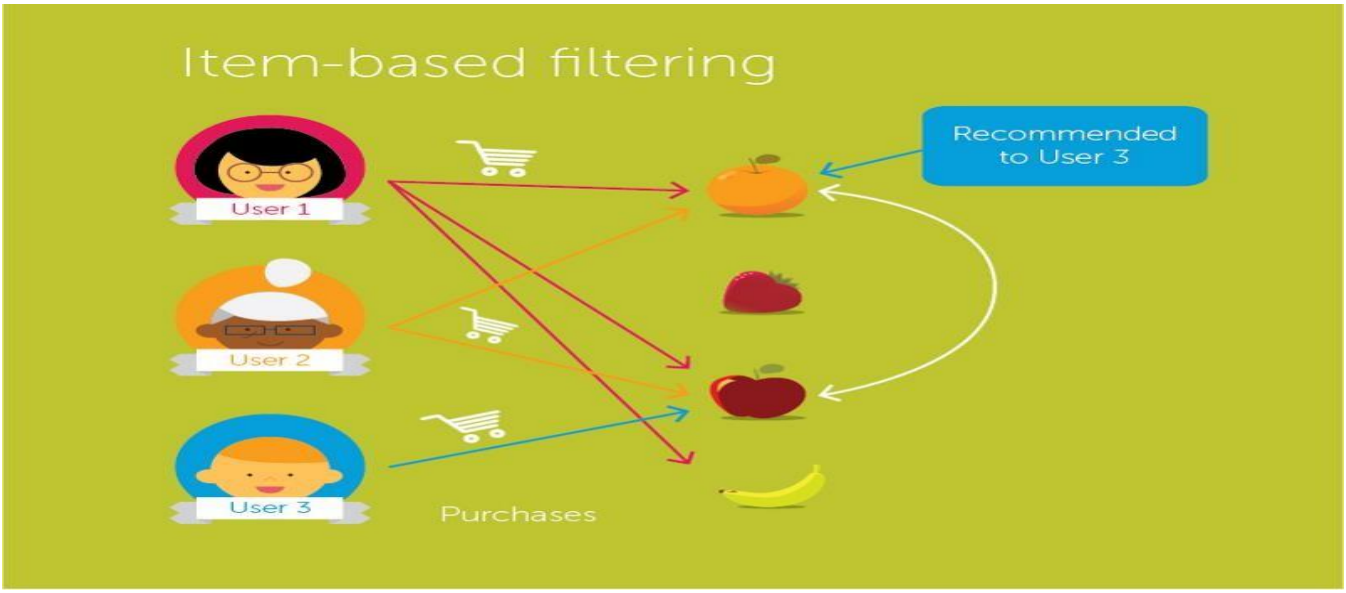
2.2.2 Collaborative filtering recommendation systems :

Collaborative filtering system items based on similarity measures between users and/or items. The basic methodology of collaborative filtering systems is that these undetermined ratings can be credited since the noticed ratings are often highly linked across several users and items This is based on the scenario where a person asks his friends, who have similar tastes, to recommend him some movies.

Advantages of Collaborative filtering system is that It is dependent on the relation between users, which implies that it is content-independent.

Many algorithms have been used in measuring user similarity or item similarity in recommender systems — for example, the k-nearest neighbour approach and the Pearson Correlation.

Image 2.2- Collaborative filtering recommendation



2.3 Applications of Recommendation Systems:

- Product Recommendations
- Movie Recommendations
- News Articles

Python Library used:

- **Pandas: -** It is a perfect tool for data wrangling process. It is designed for quick and easy data manipulation, aggregation and visualization.
- **Math: -** This module provides access to the mathematical functions defined by the C standard.
- **Numpy: -** Numpy stands for Numerical Python, provides an abundance of useful features for operations on n-arrays and matrices in python.
- **Matplotlib: -** It enables you to make different types of chart. It mainly used for data visualization process.

CHAPTER – 3

IMPLEMENTATION

Now we will implement Content-based recommendation for movies by using Python and the Pandas library. There are three main steps in the working of the project –

I. Acquiring the data
II. Preprocessing
III. Content-Based Filtering

I. Acquiring the Data :

As I discussed earlier, the main data sets is taken from the MovieLens. These data sets are csv format files.

- **Loading the data:**

  This is the first step for data modelling. In this step load the movies dataset using the pandas library. Here the pandas library is used for reading the CSV file by read_csv function.

- **Find the attributes and their types :**

  By using the head function, we find the top five values of the data in which all attribute values of the corresponding data are given. Then by using the info function, we describe the class type, the no of data in the dataset and type of attributes. And last we used to describe function to describe mean, standard deviation, minimum value, the maximum value of each attribute.

Table 3.1- Top five value of movies dataset

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Table 3.2- Information about movies dataset

```
<class 'pandas.core.frame.DataFrame'
RangeIndex: 9742 entries, 0 to 9741
Data columns (total 3 columns):
movieId    9742 non-null int64
title      9742 non-null object
genres     9742 non-null object
dtypes: int64(1), object(2)
memory usage: 228.5+ KB
```

II. Preprocessing of the data :

- **Removing year from ‘title‘ column :**

Using regular expressions in movies dataset, we find a year stored between parentheses with the movie name in 'title' column. We don't conflict with movies that have years in their titles. So we are going to remove the parentheses and also removing the years from the 'title' column and placed it into a new column and give this column name as 'year'. By applying the strip function to get rid of any ending whitespace characters that may have appeared in 'title' column.

- **Splitting the values in the ' genre' :**

Split the values in the 'genres' column into a list of genres to simplify future use by applying Python's split string function on the genre column.

Table 3.3- Top five value of movies dataset after removing year from the titile of the movie and splitting the value in 'genre'

	movieId	title	genres	year
0	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995
1	2	Jumanji	[Adventure, Children, Fantasy]	1995
2	3	Grumpier Old Men	[Comedy, Romance]	1995
3	4	Waiting to Exhale	[Comedy, Drama, Romance]	1995
4	5	Father of the Bride Part II	[Comedy]	1995

- **Converting the list of genres to a vector :**

Since keeping genres in a list format is not optimal for the content-based recommendation system technique, we will convert the list of genres to a vector where each column corresponds to one possible value of the feature. This encoding is needed for feeding categorical data. In this case, we store every different genre in columns that contain either 1 or 0. 1 shows that a movie has that genre, and 0 shows that it doesn't.

➢ **Table 3.4- This table shows the vector form of the genres**

	movieId	title	genres	year	Adventure	Animation	Children	Comedy	Fantasy	Romance	...	Horror	Mystery	Sci-Fi	War	Musical	Documentar
0	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995	1.0	1.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0
1	2	Jumanji	[Adventure, Children, Fantasy]	1995	1.0	0.0	1.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0
257	296	Pulp Fiction	[Comedy, Crime, Drama, Thriller]	1994	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0
973	1274	Akira	[Action, Adventure, Animation, Sci-Fi]	1988	1.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0
1445	1968	Breakfast Club, The	[Comedy, Drama]	1985	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0

5 rows × 24 columns

III Content-Based Filtering:

This technique attempts to figure out what a user's favourite aspects of an item is and then recommends items that present those aspects. In our case, we are going to try to figure out the input's favourite genres from the movies and ratings given

This filtering process occurred in following steps:

Step 1:- Creating an input user to recommend movies :

Add movies from the user's profile and their respective ratings given by the user in userInput. We are free to add more and more movie to get the best recommendation. Just be sure to write it in with capital letters and if a movie starts with a "The", like "The Matrix" then write it in like this: 'Matrix, The'.

Table 3.5- This table shows data from the user's profile

	title	rating
0	Breakfast Club, The	5.0
1	Toy Story	3.5
2	Jumanji	2.0
3	Pulp Fiction	5.0
4	Akira	4.5

Step 2 :-Add movieId to input user :

With the input complete, I extract the input movies' ID's from the movies data frame and add them into the input data frame. We have achieved this by first filtering out the rows that contain the input movies' title and then merging this subset with the input data frame. We also drop unnecessary columns for the input to save memory space.

Table 3.6- This table shows data from the user's profile after adding movieId

	movieId	title	rating
0	1	Toy Story	3.5
1	2	Jumanji	2.0
2	296	Pulp Fiction	5.0
3	1274	Akira	4.5
4	1968	Breakfast Club, The	5.0

Step 3 :- Learning the input preferenes:

For learning the input preferences, we get the subset of movies that the input has watched from the Dataframe containing genres defined with binary values. Now we have only needed the actual genre table, so clean this by resetting the index and dropping the movieId, title, genres and year columns.

Now we start learning the input's preferences. To do this, we turn each genre into weights. We have done this by using the input's reviews and multiplying them into the input's genre table and then summing up the resulting table by column. This operation is a dot product between a matrix and a vector, so we have done this by calling Pandas' "dot" function. Now, we have the weights for every of the user's preferences. This is known as the ***User Profile.*** Using this, we can recommend movies that satisfy the user's preferences.

Table 3.7- This table shows the subset of movies datset that the input has watched from the Dataframe containing genres with binary values

	movieId	title	genres	year	Adventure	Animation	Children	Comedy	Fantasy	Romance	...	Horror	Mystery	Sci-Fi	War	Musical	Documentar
0	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995	1.0	1.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0
1	2	Jumanji	[Adventure, Children, Fantasy]	1995	1.0	0.0	1.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0
257	296	Pulp Fiction	[Comedy, Crime, Drama, Thriller]	1994	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0
973	1274	Akira	[Action, Adventure, Animation, Sci-Fi]	1988	1.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0
1445	1968	Breakfast Club, The	[Comedy, Drama]	1985	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0

5 rows × 24 columns

Table 3.8- This table shows the table 3.5 after removing 'movieId', 'title', 'genres', and 'year' attribute

	Adventure	Animation	Children	Comedy	Fantasy	Romance	Drama	Action	Crime	Thriller	Horror	Mystery	Sci-Fi	War	Musical	Documentary	IMAX	W
0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Table 3.9- This table shows the rating column of user's data

```
0    3.5
1    2.0
2    5.0
3    4.5
4    5.0
Name: rating, dtype: float64
```

Now the dot product between Transpose of matrix elements of table 3.6 and matrix elements of table3.7 then we get user's profile.

Table 3.10- This table shows the user's profile

```
Adventure              10.0
Animation               8.0
Children                5.5
Comedy                 13.5
Fantasy                 5.5
Romance                 0.0
Drama                  10.0
Action                  4.5
Crime                   5.0
Thriller                5.0
Horror                  0.0
Mystery                 0.0
Sci-Fi                  4.5
War                     0.0
Musical                 0.0
Documentary             0.0
IMAX                    0.0
Western                 0.0
Film-Noir               0.0
(no genres listed)      0.0
dtype: float64
```

Step 4: Creating the list of top 20 recommended movie table :

We have started by extracting the genre table from the original data frame, .and after that drop the unnecessary information. Now we have With the input's profile and the complete list of movies and their genres. After that, we're going to multiply the genres by the weights and then take the weighted average of every movie based on the input profile and sort it in descending order then we have found the recommendation table as a result of the top 20 movies that ***most satisfy it.***

CHAPTER – 4

RESULTS AND DISCUSSION

After the implementation of the necessary steps as discussed above. Using term document weights, our system is ableto recommend the following movies as shown in fig(). Our system has recommended these movies based on the user's choice of the movies. We have used the dataset of the movies watched by a user, in this we are using the movie names and its ratings given by the user according to his analysis of that movie. After performing necessary algorithms on user's dataset our system is recommending a list of best suited 15 movies for that

Table 4.1- This table show the list of 15 recommended movies as a result of this project

	movieId	title	genres	year
559	673	Space Jam	[Adventure, Animation, Children, Comedy, Fanta...	1996
1390	1907	Mulan	[Adventure, Animation, Children, Comedy, Drama...	1998
2250	2987	Who Framed Roger Rabbit?	[Adventure, Animation, Children, Comedy, Crime...	1988
4631	6902	Interstate 60	[Adventure, Comedy, Drama, Fantasy, Mystery, S...	2002
5490	26340	Twelve Tasks of Asterix, The (Les douze travau...	[Action, Adventure, Animation, Children, Comed...	1976
5819	32031	Robots	[Adventure, Animation, Children, Comedy, Fanta...	2005
6448	51939	TMNT (Teenage Mutant Ninja Turtles)	[Action, Adventure, Animation, Children, Comed...	2007
7441	81132	Rubber	[Action, Adventure, Comedy, Crime, Drama, Film...	2010
7550	85261	Mars Needs Moms	[Action, Adventure, Animation, Children, Comed...	2011
8349	108540	Ernest & Célestine (Ernest et Célestine)	[Adventure, Animation, Children, Comedy, Drama...	2012
8357	108932	The Lego Movie	[Action, Adventure, Animation, Children, Comed...	2014
8597	117646	Dragonheart 2: A New Beginning	[Action, Adventure, Comedy, Drama, Fantasy, Th...	2000
8806	130520	Home	[Adventure, Animation, Children, Comedy, Fanta...	2015
8900	134853	Inside Out	[Adventure, Animation, Children, Comedy, Drama...	2015
9169	148775	Wizards of Waverly Place: The Movie	[Adventure, Children, Comedy, Drama, Fantasy, ...	2009

CHAPTER – 6

CONCLUSION AND FUTURE WORK

In my project, content-based filtering algorithm is used to give the best recommendation of movies to the user. It solves the new item recommendation problem and gives an idea about the current trends of popular movies and users interests. This is very helpful for a movie producer to plan new movies. But this method can be expanded by including more criteria to help in categorization the movies. The most obvious ideas are to add features to suggest movies with common actors, directors or writers. In addition, we could try to develop hybrid methods that try to combine the advantages of both content-based methods and collaborative filtering into one recommendation system.

BIBLIOGRAPHY

1 Ali, K., van Stam, W.: TiVo: Making Show Recommendations Using a Distributed Collaborative Filtering Architecture. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, pp. 394–401. ACM Press, New York (2004)

2 Balabanovic, M., Shoham, Y.: FAB: Content-based, Collaborative Recommendation Communications of the Association for Computing Machinery 40(3), 66–72 (1997)

3 Basu, C., Hirsh, H., Cohen, W.: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In: Proceedings of the 15th National Conference on Artificial Intelligence, Madison, Wisconsin, pp. 714–720 (1998)

4 Belkin, N., Croft, B.: Information Filtering and Information Retrieval: Two Sides of the Same Coin? Communications of the ACM 35(12), 29–38 (1992)

5 Burke, R.: Hybrid Web Recommender Systems. In: Brusilovsky, P., Kobsa, A. Nejdl, W.(eds.) The Adaptive Web: Methods and Strategies of Web Personalization. LNCS,vol. 4321, pp. 377–408. Springer, Heidelberg (2007)

6 Kotsiantis, Sotiris. (2007). Supervised Machine Learning: A Review of Classification Techniques. Informatica (Ljubljana). 31.

7 Manjula, R & Jain, Shubham & Srivastava, Sharad & Kher, Pranav. (2017). Real estate value prediction using multivariate regression models. IOP Conference Series: Materials Science and Engineering. 263. 042098. 10.1088/1757-899X/263/4/042098.

8 Sathya, R. & Abraham, Annamma. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. International Journal of Advanced Research in Artificial Intelligence. 2. 10.14569/IJARAI.2013.020206

9 https://link.springer.com/chapter/10.1007/978-3-540-72079-9_10

10 http://infolab.stanford.edu/~ullman/mmds/ch9.pdf

11 http://cs229.stanford.edu/proj2018/report/128.pdf

12 http://www.ijesrt.com/issues%20pdf%20file/Archive-2016/November-2016/63.pdf

13 https://en.wikipedia.org/wiki/Recommender_system

14 https://towardsdatascience.com/brief-on-recommender-systems-b86a1068a4dd

15 https://indatalabs.com/blog/big-data-behind-recommender-systems

16 https://docs.microsoft.com/en-us/analysis-services/data-mining/training-and-testing-data-sets

17 https://mindmajix.com/polynomial-regression

18 https://towardsdatascience.com/https-medium-com-lorrli-classific

19 http://cdn2.content.compendiumblog.com/uploads/user/458939f4-fe08-4dbc-b271-efca0f5a2682/ba6a552e-3bc0-4eed-9c9a-eae3ade49498/Image/babb38258faeb3eeffcf297614c3a2db/r2_formula_ss_total.jpg

20 https://www.includehelp.com/ml-ai/Images/rmse-.jpg