International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN (online): 2347-5552, Volume-11, Issue-2, March 2023
https://doi.org/10.55524/ijircst.2023.11.2.6
Article ID IJIRD-1237, Pages 28-33
www.ijircst.org

Development of a Movie Recommendation System - MoviepleX

Vaani Gupta[1], **and Khushboo Tripathi**[2]

[1,2]Department of Computer Science and Engineering, Amity University, Gurgaon, Haryana, India

Correspondence should be addressed to Vaani Gupta; vaanigupta01@gmail.com

ABSTRACT- The content recommendation model, "Development of a Movie Recommendation System - MoviepleX" is aimed at providing accurate movie recommendations to users, on the basis of similarity with the movie they would enter for reference, using machine learning algorithms, functions and metrics. It is built using the tmdb_5000 dataset, taken from Kaggle. The data consists of a number of features like cast, crew, genre, budget, overview, runtime, tagline, popularity, production unit and revenue corresponding to 4803 Hollywood movies that are a part of the tmdb database.

Recommendation engines are a subclass of information filtering systems that seek to predict the 'rating' or 'preference' a user would give to an item, a movie in case of a movie recommender. Streaming media services like Netflix & Disney+ Hotstar employ highly efficient content recommendation systems, which can play a huge role as game-changers in a streaming service's success or failure. These content-based recommenders are what keep our entertainment rhythm going, serving us the best material out there, based on our own personal interests, choices, likes & dislikes. Movie recommendation systems provide a mechanism to assist viewers and subscribers of streaming platforms by classifying movies based on similar interests of users. A movie recommendation is important in our social life due to its strength in providing enhanced entertainment.

The model proposed in this paper uses machine learning's capability to identify patterns and build prediction and recommendation mechanisms using provided data. A machine learning web application was created for the recommendation engine, which was deployed onto Heroku, a container-based cloud Platform as a Service (PaaS), used to deploy, manage, and scale modern apps. The app deployment was made through Streamlit. By having a webpage for the ML - application, it has been made accessible and beneficial to public.

KEYWORDS- Streaming Media, Movie Recommendation, Machine Learning, Heroku

I. INTRODUCTION

In today's digital world where there is an endless variety of content to be consumed in the form of e-books, videos, blogs, vlogs, webseries, movies, etc. [15,18], finding the content of one's liking has become an irksome task [8,11]. With the popularity of streaming services like Netflix growing at a neck-breaking speed, especially since the worldwide lockdown, people are engaging themselves by watching shows, web-series and movies more than ever. Digital content providers want to engage maximum users on their service for as long as they can. [21] This is where the movie recommender system developed in this paper can come handy, where the content providers recommend users the movie-content according to the users' interests and choices in films.

After all, the raging popularity of top platforms like Netflix, Prime Video & YouTube can largely be attributed to their highly efficient & accurate content recommendation systems. In this paper, a movie recommendation engine named MoviepleX is proposed to tackle the challenge of attracting as many viewers & subscribers as possible.

A. Purpose and User's Role

MoviepleX designed for this paper, works based on the machine learning model created to suggest five movies to its users based on a movie they like, by checking for the users' interest in various elements of the movie, like:

- Genre
- Plot and Overview
- Starcast
- Director, and other features.

On the webpage that allows a user to access the recommendation application, the user is simply required to choose from a drop down menu or enter the name of a movie they'd like to see more movies like, and voila! They would be presented with 5 similar movies with their posters & brief overview for reference. It's as simple as that to use.

B. A Suite of Python Modules used

Python programming language is used to design the model in *Jupyter Notebook* [24-25]. Several extensive Python Libraries were used to bring this work to life, listed as follows:

- ast
- nltk
- PorterStemmer
- CountVectorizer
- cosine_similarity
- Pickle

II. BACKGROUND AND LITERATURE REVIEW

Content-based Filtering Recommendation Engines suggest recommendations based on the item metadata [9], the main

idea being that if a user likes an item, then he or she will also like items similar to it. The methodology taken up to develop the algorithm which compares the similarity between two movies can be explained using an example of *cosine distances*. The cosine distance between the vectors of the item and the user can be used to determine its preference to the user. The vector for a user would have a positive number for actors that tend to appear in movies the user likes and negative numbers for actors the user doesn't like.

A. Literature Review

Recommender systems have become an important research field since the emergence of the first paper on collaborative filtering [10,12,16] in the mid-1990s. In the earlier approaches, it was mainly AI, ML and Data Mining which was used to create such systems. Research in movie recommenders has followed the principle that a recommendation method should be capable of noticing preferences involuntarily to make playlists accordingly. Most articles and papers published have implemented the data mining techniques of K-nearest neighbor algorithm, clustering and association rules [17], as can be seen below:

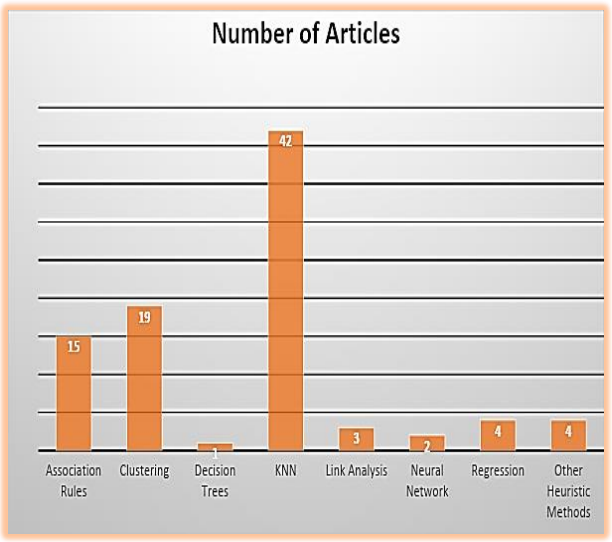

Figure 1: Distribution of Articles by Data Mining Techniques

Author & Reference	Approach and Techniques Used
Kim Mucheol et al. [1]	i. Interactive movie recommendation engine that constructs adapted suggestion of films in online communities ii. Develops the grouping conscious community network model capable of limiting dynamics of socially mediated details communicated in communal networks
Colombo et al. [2]	i. RecomMetz - a mobile recommender dependent on context-aware knowledge ii. Uses Semantic Web technologies – competent & effective iii. In free time domain only for movie show times
Fernandez et al. [3]	i. Slope One algorithm for calculating the specific prophecy ii. Multiplicative Utilitarian Strategy as a replica to give suggestions to a whole crowd of cinema goers
Symeonidis et al. [4]	i. MoviExplain' - perfect and justifiable recommendations ii. Allows user to verify reasoning behind a recommendation
Christakou et al. [5]	i. semi-supervised learning dependent clustering technique ii. merges collaborative and content - based information iii. checked on Movie Lens DS, high precision
Rattanajit et al. [6]	i. Naive Bayes approach for doing pseudo ratings reliant on categorized multi criteria of client preferences; accurate ii. multi regression used to examine appropriate information of client to include multiple dimensions iii. created on a movie domain called Modernize Movie
Nanou et al. [7]	i. problems associated with movie recommendations ii. survey of former techniques, different methods compared; other popular recommenders focused on user opinion and approval iii. most effective method - "planned outline", "textbook & videotape" interfaces iv. strong constructive association originated between client opinion and approval
Luis M Capos et al. [19]	i. both content based and collaborative filtering have their own drawbacks ii. proposed new system - combination of Bayesian network & collaborative filtering iii. optimized for given problem, provides probability distributions for useful inferences
Harpreet Kaur et al. [20]	i. hybrid system - mix of content and collaborative filtering algorithm ii. context of the movies; user - user as well as user - item relationship considered

From clustering techniques dependent on semi - supervised learning to semantic web technologies and Naive Bayes, there has been a plethora of different techniques people have applied in establishing a model to recommend movies, based on content filtering as well as collaborative filtering. Inspite of the presence of examples on the internet based on the cosine_similarity technique used in this paper, no past works using such an approach were covered in this literature survey.

Hence, it can be concluded that the research conducted on movie recommendation engines so far yields that they pose the most effective solution addressing the rising problem of data and information overload. They facilitate choice-making by saving time and energy.

Prospects in this field may focus on enhancement of the existing models, methods, techniques and algorithms used so that the recommendation systems' predictions and quality can be improved.

III. METHODOLOGY AND DESIGN

The content-based movie recommender developed in this paper works with the movie preference data provided by the user. The basic design blueprint of the entire process which took place behind the scenes while developing the recommendation engine can be seen in Fig. 3.1 [28-29]. The machine learning model compares the entered movie's characteristics to other movies' in the data base. Then it outputs the names of the five movies with genre, actors, overview, etc. that are most similar to the movie entered by the user.

The overall scope of the work involved in building the ML-powered recommender can be listed out as follows:

- Understanding the problem and final goal
- Dataset collection
- Data preparation and pre-processing
- Vectorising the important key word determinants in the data
- Measuring the angle between the vectors to check for similarity

A. Data Pre-Processing

1)Data Filtering and Noise Removal

It is very crucial to make the data useful because unwanted or null values (NaN) can cause unsatisfactory results or may lead to lower accuracy. Necessary operations need to be performed on the DataFrame to replace null or missing values with default values or to remove rows consisting absent values. Data filtering and removal of unnecessary variables was performed for the datasets which were first merged into a single dataset. It further underwent suitable type conversions, using methods like literal_eval.

2)Feature Selection & Engineering

Feature selection is a crucial step for selecting the required elements from the data set which enable improved performance measurement. Thus, having only significant features and reducing the number of irrelevant attributes and values increases the accuracy of recommendation. It was applied on this work's data using the max_features parameter while defining the CountVectorizer function.

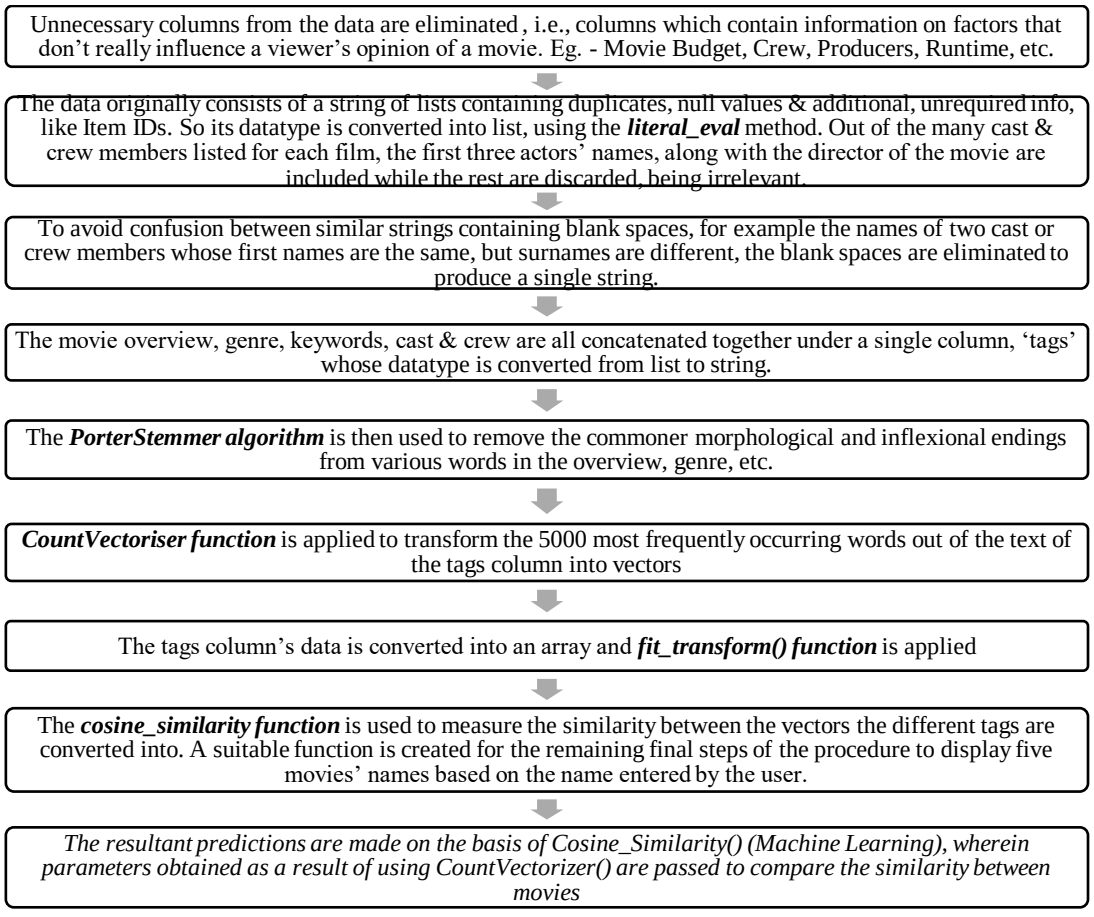

Figure 2: Methodology Blueprint Flowchart

B. Data Scaling - Fitting and Transformation

Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained. A model that is well-fitted produces more accurate outcomes. The fit_transform() function has been used on the tags column data to scale it. Here, the model will take the 5000 most relevant or say, important words of the column containing info on overview, genre, cast, etc. of the movie and use stop_words of English.

IV. ARITHMETIC USED – COSINE SIMILARITY

Vectorization was applied on the text of the tags column, containing key information on every movie, like its genre, overview, cast and director [26-27]. It was converted into numerical vectors (in the form of a sparse matrix) using CountVectorizer function, to make the task of comparison between two movies easier. To measure the cosine angle between every vector, the cosine_similarity function was used.

Cosine similarity is a measure of similarity, often used to measure document similarity in text analysis. The following formula is used to compute it:

$$Similarity = (A.B) / (||A||.||B||)$$

where A and B are vectors.

- A.B is dot product of A and B: computed as the sum of element-wise product of A and B
- ||A|| is L2 norm of A: computed as the square root of the sum of squares of elements of vector A

Euclidean (L2) normalization projects the vectors onto the unit sphere, and their dot product is then the cosine of the angle between the points denoted by the vectors.

Each item is stored as a vector of its attributes in an n-dimensional space [14]. The value of cosine will increase with decreasing value of the angle which signifies more similarity. Vectors are length normalized (of length 1) and then the cosine calculation is simply the sum-product of vectors.

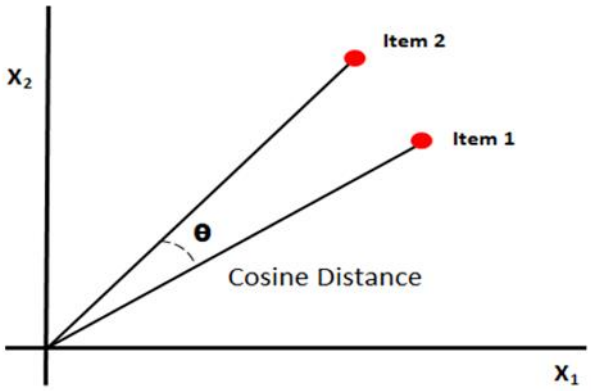

Figure 3: Diagrammatic Representation of Cosine Similarity

V. IMPLEMENTATION AND RESULTS DISCUSSION

Experimentation on the dataset was done by merging & concatenation of datasets, columns and attributes values.

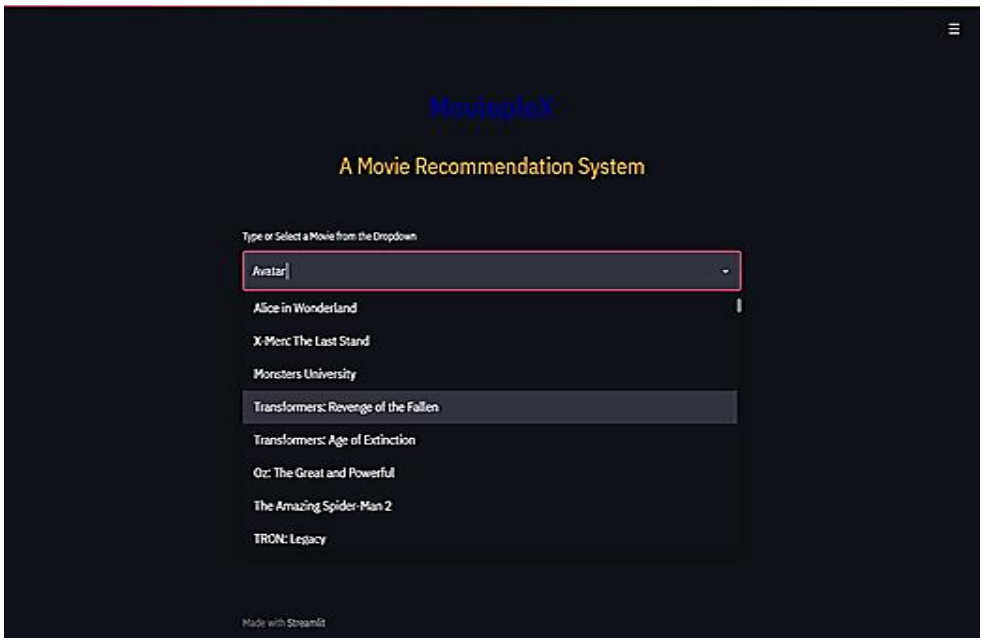

Figure 4: Streamlit Application - Screenshot 1

Unrequired attributes like production unit, runtime, budget, spoken languages, word count, ID, homepage, etc., were removed from the dataset for more accurate details [13] and to reduce computation load and time. The movies recommended by the engine are similar to the movie entered for reference. Hence, the results are accurate.

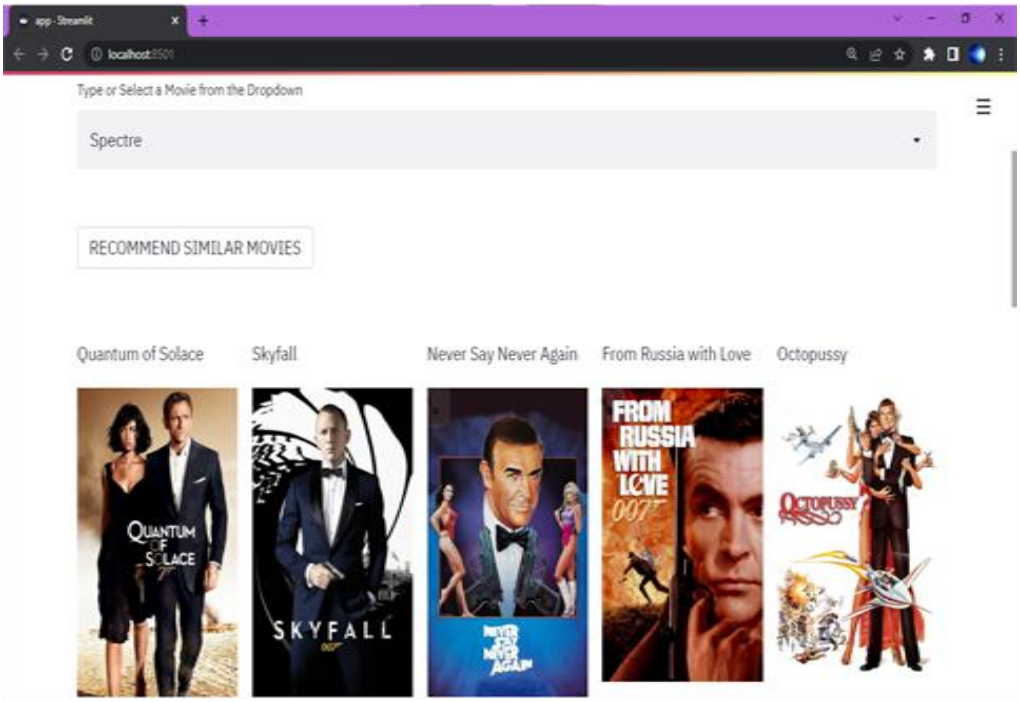

Figure 5: Streamlit Application - Screenshot 2

After designing the recommendation engine, the pickle module of python was used to save the serialized ML model to disk. It was then loaded in the app.py file. A webpage app layout was designed using Streamlit.

VI. CONCLUSION AND FUTURE WORK

The cosine similarity method used to measure the similarity between movies is a novel and unique method to create the algorithm of a recommender as seen in research on past works. MoviepleX proposed in this paper requires no user information, subscriptions or any other data, other than the name of a movie the user would like to watch another movie like – that's where this paper can prove to outshine its counterparts in the existing system based off user info.

This paper can be carried further to next steps of working with stronger performance & enhanced accuracy by creating the right mix of ML & DL [22,23]. Given how simple and straightforward the programming is, there should be no problem studying it and taking it to the next level in prospects.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest

REFERENCES

[1] Kim, Mucheol, and S. O. Park, "Group affinity based social trust model for an intelligent movie recommender system", Multimedia tools and applications 64, vol. no. 2, pp. 505-516, 2013.

[2] Colombo, Mendoza, L. Omar, R. V. García, A. R. González, G.A. Hernández, and J. J. S. Zapater, "RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes", Expert Systems with Applications 42, vol. no. 3, pp. 1202-1222, 2015.

[3] Fernandez, George, W. Lopez, F. Olivera, B. Rienzi, and P. R. Bocca, "Let's go to the cinema!", a movie recommender system for ephemeral groups of users", Computing Conference (CLEI), XL Latin American, IEEE, pp. 1-12, 2014

[4] Symeonidis, Panagiotis, A. Nanopoulos, and Y. Manolopoulos, "MoviExplain: a recommender system with explanations", Third ACM conference on Recommender systems, pp. 317-320, 2009

[5] Christakou, Christina, L. Lefakis, S. Vrettos, and A. Stafylopatis, "A movie recommender system based on semi-supervised clustering", Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on IEEE, vol. 2, pp.897-903, 2005

[6] R. Banjong, Nutcha, and S. Maneeroj, "Multi criteria pseudo rating and multidimensional user profile for movie recommender system", Computer Science and Information Technology (ICCSIT) 2009, 2nd IEEE International Conference on IEEE, pp. 596-601, 2009

[7] Nanou, Theodora, G. Lekakos, and K. Fouskas, "The effects of recommendations" presentation on persuasion and satisfaction in a movie recommender system", Multimedia systems 16, vol. no. 4-5, 219-230, 2010

[8] E. Rich, "User modeling via stereotypes", Cognitive Science, Vol. 3, No. 4, pp. 329–354, 1979.

[9] M. Sanderson, and W. B. Croft, "The History of Information Retrieval Research, Proceedings of the IEEE, Vol. 100, pp. 1444–1451, 2012.

[10] P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens: An open architechure for collaborative filtering of netnews", In Proceedings of the ACM Conf. Computer Support Cooperative Work (CSC), pp. 175-186, 1994.

[11] R. E. Nisbett, and T. D. Wilson, "Telling more than we can know: Verbal reports on mental processes", Psychological Review, Vol. 84, No. 3, pp. 231-259, 1977.
[12] D. Goldberg, B. Oki, D. Nichols, and D. B. Terry, "Using Collaborative Filtering to Weave an Information Tapestry", Communications of the ACM, December, Vol. 35, No. 12, pp. 61-70, 1992.
[13] Online available: www.youtube.com/
[14] G. Salton, A. Wong, C. S. Yang, "A vector space model for automatic indexing", Communications of the ACM, Vol.18, No.11, pp. 613-620, 1975.
[15] M.H. Ferrara, M. P. LaMeau, "Pandora Radio/Music Genome Project. Innovation Masters: History's Best Examples of Business Transformation. Detroit", Gale Virtual Reference Library, pp. 267-270, 2012.
[16] M. Balabanovic´, Y Shoham, "Fab: Content-based, Collaborative Recommendation", Communications of the ACM, Vol.40, No.3, pp.66-72, 1997.
[17] "Recommendation-system", en.citizendium.org/wiki/Recommendation_system
[18] J. B. Schafer, J. A. Konstan, J. Riedl, "E-Commerce recommendation applications", Data Mining and Knowledge Discovery, Vol. 5, No. 1, pp. 115–153, 2001.
[19] L. M. de Campos, J. M. F. Luna, J.F. Huete, M. A. R. Morales; "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks", International Journal of Approximate Reasoning, revised 2010.
[20] H. K. Virk, Er. M. Singh, "Analysis and Design of Hybrid Online Movie Recommender System", International Journal of Innovations in Engineering and Technology (IJIET), Vol. 5, Issue 2, 2015
[21] http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html?_r=0
[22] Y. Himeur, A. Sayed, A. Alsalemi, F. Bensaali, A. Amira, I. Varlamis, M. Eirinaki, C. Sardianos, G. Dimitrakopoulos. "Blockchain-based recommender systems: Applications, challenges and future opportunities." Comput. Sci. Rev., 100439, p. 43, 2022.
[23] S. Jayalakshmi, N. Ganesh, R. Čep, J. Senthil Murugan. "Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions". Sensors 22, no. 13: 4904, 2022. https://doi.org/10.3390/s22134904.
[24] "Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido
[25] "Machine Learning Mastery with Python" by Jason Brownlee
[26] Online available: www.towardsdatascience.com/
[27] Online available: www.researchgate.net/
[28] Online available: www.ijesrt.com/
[29] Online available: stackoverflow.com