IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 11, November 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 11, November 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0911035 |

Movie Recommendation Engine Using Machine Learning

Tushar Chawla, Ajay Kaushik

B.Tech Student, Dept. of I.T., Maharaja Agrasen Institute of Technology, New Delhi, India

Assistant Professor, Dept. of I.T., Maharaja Agrasen Institute of Technology, New Delhi, India

ABSTRACT: This paper examines about suggestions of the motion pictures. A film suggestion is significant in our public activity because of its solidarity in giving upgraded diversion. Such a framework can propose a bunch of motion pictures to clients dependent on their advantage, or the popularities of the films. A proposal framework is utilized to recommend things to buy or to see. They direct clients towards those things which can address their issues through chopping down huge data set of Information. A recommender framework, or a suggestion framework (now and then supplanting 'framework' with an equivalent like stage or motor), is a subclass of data sifting framework that tries to anticipate the "rating" or "inclination" a client would provide for a thing. They are principally utilized in business applications, likewise assist clients with finding the motion pictures of their decisions dependent on the film insight of different clients in proficient and compelling way without burning through much time in pointless perusing.

KEYWORDS: TF-IDF Vectorizer; Cosine Similarity; Data Visualization

I. INTRODUCTION

Recommender frameworks normally utilize either or both collaborative based and content-based separating (otherwise called the character based methodology), just as different frameworks for example, information based frameworks. Community oriented sifting approaches fabricate a model from a client's past conduct (things recently bought or chose as well as mathematical appraisals given to those things) just as comparable choices made by different clients. This model is then used to anticipate things (or appraisals for things) that the client might have an interest in. Content-based separating approaches use a progression of discrete, pre-labeled attributes of a thing to suggest extra things with comparative properties. Current recommender frameworks ordinarily join at least one methodologies into a half breed framework. The contrasts among synergistic and content-based sifting can be exhibited by looking at two early music recommender frameworks – Last. FM and the application of Pandora which is also a radio streaming service.

1. Last FM - is a streaming service which experts in radio department and is one of the first to implement recommender frameworks and what it prefers is the collaborative mode of the frameworks where each and every recommendation is decided by what the user prefers to listen and then makes a list of categories for what the similar types of users also prefers to hear. It is usually defined on the behavior of the users and categorize what set of genres are preferred by a particular band of users.

2. Pandora – is again a streaming service application which is one of the first to use content based frameworks. What happens in a content based framework is that the recommendations are made in accordance to preference of users with respect to genres or artists so it doesn't ideally requires huge amounts of data to give accurate recommendations as it has to decide on the basis of existing content.
As we now if there are positives there are bound to be negatives for both systems. In the above model, Last. Fm as it requires or needs huge amounts of data for its suggestions it is sometimes slow comparable to other frameworks There also exists a cold start problem for the collaborative based framework. Pandora, however needs less amounts of data so generally requires much less time still it is much less elaborative as it can make recommendations which are based on the only existing content like the number of music albums or songs present in the datasets Recommender structures are a useful decision to look computations since they help customers with finding things they may not have considered to be regardless. Of note, recommender systems are routinely executed using web search apparatuses requesting non-regular data.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 11, November 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0911035 |

II. METHODOLOGY

ALGORITHMS FOR OUR RECOMMENDATION SYSTEM

· TF-IDF VECTORIZER · COSINE SIMILARITY 1. TF IDF VECTORIZER:

TF-IDF (Term Frequency and Inverse Document Frequency) is a factual measure that assesses how pertinent a word is to a record in an assortment of archives. This is finished by increasing two measurements: how frequently a word shows up in a report, and the reverse archive recurrence of the word across a bunch of records. It has many utilizations, above all in robotized text investigation, and is exceptionally valuable for scoring words in AI calculations for Natural Language Processing (NLP).

TF-IDF was imagined for record search and data recovery. It works by expanding relatively to the occasions a word shows up in a record, yet is balanced by the quantity of reports that contain the word. Along these lines, words that are normal in each archive, like this, what, and if, rank low despite the fact that they might seem ordinarily, since they don't mean a lot to that report specifically.

2. COSINE SIMILARITY

A consistently used way of managing match similar reports relies upon counting the most outrageous number of typical words between the records. In any case, this system has a natural flaw. That is, as the size of the record extends, the amount of typical words will overall addition whether or not the reports talk about different subjects. The cosine equivalence beats this focal imperfection in the 'count-the-typical words' or Euclidean distance approach. Cosine likeness is an estimation used to choose how similar the records are autonomous of their size. Mathematically, it gauges the cosine of the purpose between two vectors projected during a multi-dimensional space. During this particular circumstance, the 2 vectors i'm examining are bunches containing the word counts of two reports. When plotted on a multi-dimensional space, where every estimation identifies with a word within the chronicle, the cosine likeness gets the course (the place) of the reports and not the dimensions, accepting you actually wanted the degree, register the Euclidean distance taking everything under consideration.

III. RESULT AND DISCUSSION

1. Datasets imported from kaggle are netflix_titles.csv andimdb_ratings,csv and has various fields like :

```
netflix.count()
```

```
show_id         8807
type            8807
title           8807
director        6173
cast            7982
country         7976
date_added      8797
release_year    8807
rating          8803
duration        8804
listed_in       8807
description     8807
dtype: int64
```

Figure 1: Number Of data in various fields

2. Data Visualization

```
sns.set(style='darkgrid')
ax = sns.countplot(x='type', data=netflix, palette='Set2')
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 11, November 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0911035 |

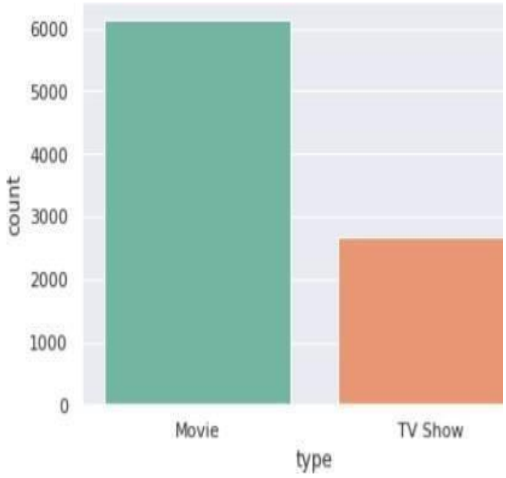


Figure 2: Comparison between Movie and TV Show

Analysis of Movie ratings:

```
plt.figure(figsize=(12,10))
sns.set(style="darkgrid")
ax = sns.countplot(x="rating", data=netflix_movies, palette="Set2", order=netflix_movies['ra
ting'].value_counts().index[0:15])
```

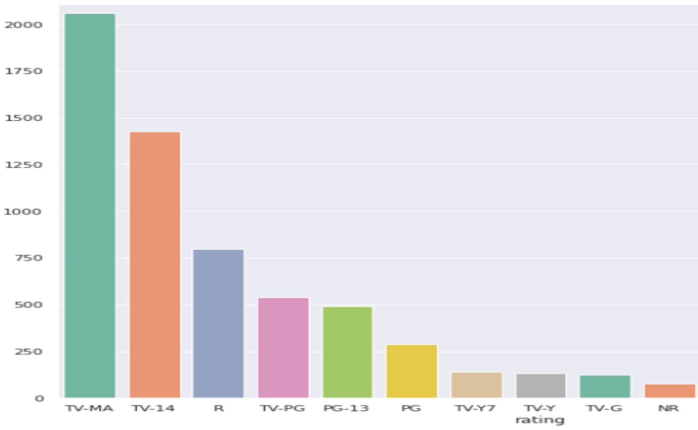


Figure 3: Here, we can see largest group of viewers are of Mature Audience and smallest group is of general "NR" movie category.

Now, performing inner join on both required datasets after dropping unnecessary fields

Analyzing top 10 rated movies on Netflix:

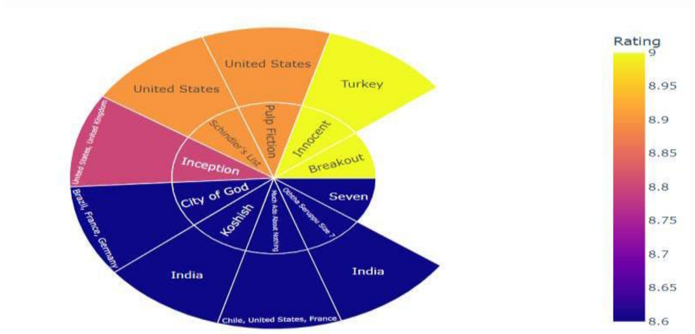
```
import plotly.express as px
top_rated = joint_data[0:10]
fig = px.sunburst(
top_rated,path=['title', 'country'],values='Rating', color='Rating')
fig.show()
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 11, November 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0911035 |

Figure 4: Top 10 Rated Movies



Year wise analysis:-

Figure 5: Here, we can see maximum no. of movies are released in 2018 Top Countries which are creating good content

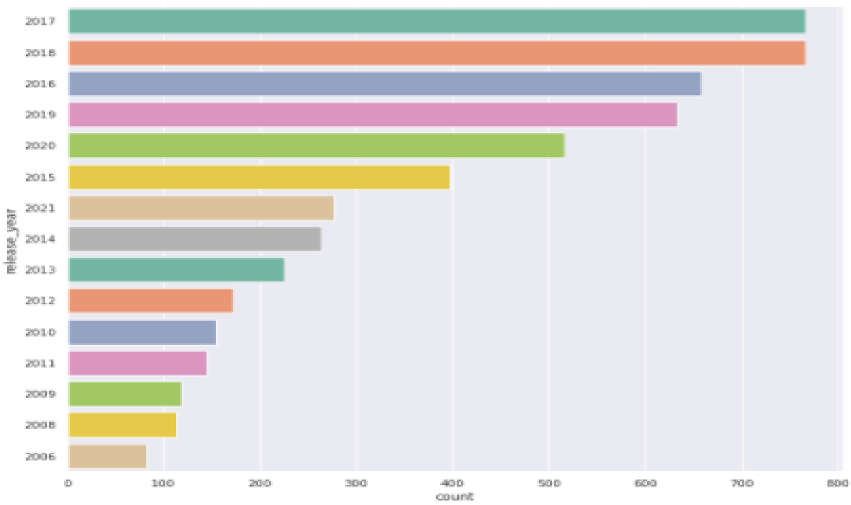


Figure 6: Movies released in terms of countries

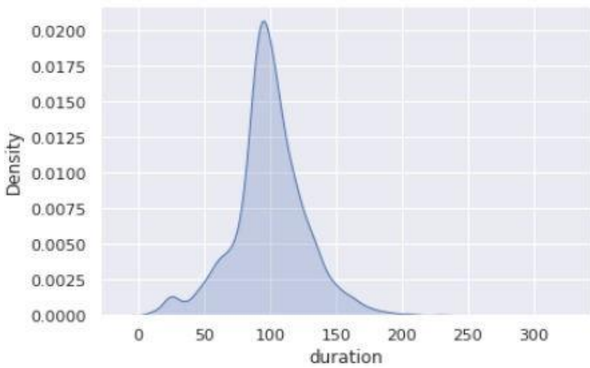


Figure 7: Here, we can see maximum movies are from 90 minutes to 120 minutes duration. Lollipop Chart for Genres in replace of Bar chart

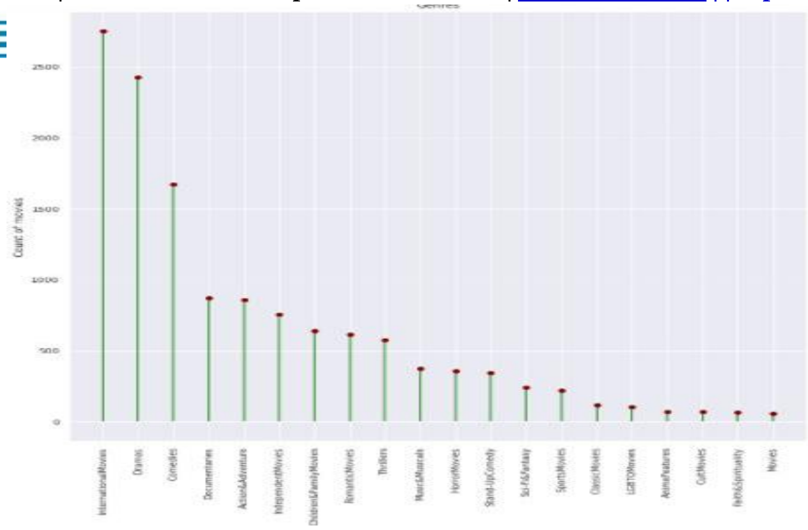


Figure 8: Here, we can see count of movies according to genres TF-IDF and Cosine Similarity and further recommendations achieved

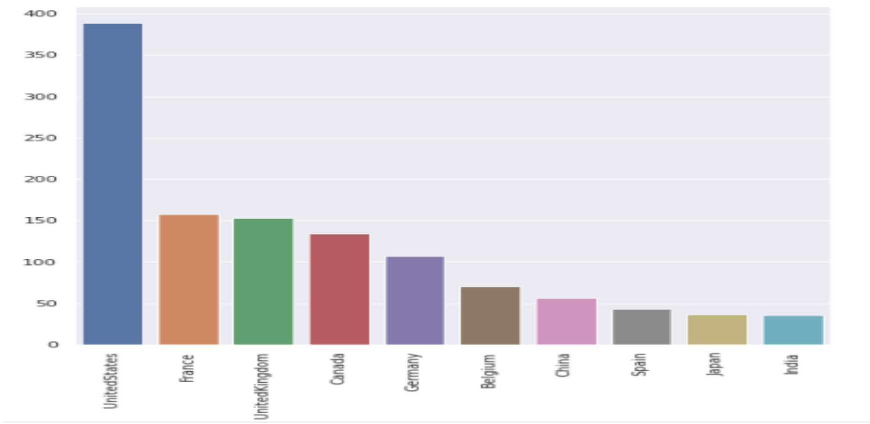


Figure 9: Top 10 recommendations achieved

IV. CONCLUSION

In this paper, we have presented Movie recommender framework for film proposal. It permits a client to choose his decisions from a given arrangement of properties and afterward suggest him a film list dependent on the aggregate load of various properties also, utilizing TF-IDF and Cosine similarity. By the idea of our framework, it is not a simple assignment to assess the exhibition since there is no Right or wrong proposal; it is simply an issue of assessments. In view of casual assessments that we completed over a little set of clients we got a positive reaction from them. We would like to have a bigger informational collection that will empower more significant results utilizing our framework. Moreover we might want to join distinctive AI and bunching calculations what's more, concentrate on the relative outcomes. Ultimately we might want to execute an online UI that has a client data set, furthermore, has the learning model custom-made to every client.

REFERENCES

[1] Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook Recommender Systems Handbook, Springer, 2011, pp. 1-35

[2] "Lead Rise of Recommendation Engines - TIME". TIME.com. 27 May 2010. Retrieved 1 June 2015.

[3] Patel, Z., Senjaliya, N., & Tejani, A. (2019). AI-enhanced optimization of heat pump sizing and design for specific applications. International Journal of Mechanical Engineering and Technology (IJMET), 10(11), 447-460.

[4] Vanetti, Marco, et al. "Content based filtering in on-line social networks." International Workshop on Privacy and Security Issues in Data Mining and Machine Learning. Springer, Berlin, Heidelberg, 2010.

[5] Van Meteren, Robin, and Maarten Van Someren. "Using content-based filtering for recommendation." Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop. Vol. 30. 2000.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 11, November 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0911035 |

[6] Schafer, J. Ben, et al. "Collaborative filtering recommender systems." The adaptive web. Springer, Berlin, Heidelberg, 2007. 291-324.
[7] Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan. Collaborative filtering recommender systems. Now Publishers Inc, 2011.
[8] M.Sabin Begum, R.Sugumar, Novel entropy based approach for cost-effective privacy preservation of intermediate datasets in cloud., Cluster Computing, The Journal of Networks, Software Tools and Application. [Springer], Volume 12, Issue 2, pp.121-132, October 2017.
[9] Rubens, Neil; Elahi, Mehdi; Sugiyama, Masashi; Kaplan, Dain (2016)."Active Learning in Recommender Systems". In Ricci, Francesco; Rokach, Lior; Shapira, Bracha (Eds.). Recommender Systems Handbook (2 ed.). Springer US.
[10] Senjaliya, N., & Tejani, A. (2020). Artificial intelligence-powered autonomous energy management system for hybrid heat pump and solar thermal integration in residential buildings. International Journal of Advanced Research in Engineering and Technology (IJARET), 11(7), 1025-1037.
[11] Karlgren, Jussi. 1990. "An Algebra for Recommendations." Syslab Working Paper 179 (1990).
[12] Karlgren, Jussi. "Newsgroup Clustering Based On User Behavior-Recommendation Algebra." SICS Research Report (1994).
[13] Karlgren, Jussi (October 2017) digital bookshelf: original work on recommender systems". Retrieved 27 October 2017.
[14] Shardanand, Upendra, and Pattie Maes. "Social information filtering: algorithms for automating “word of mouth”." In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 210-217. ACM Press/Addison-Wesley Publishing Co., 1995.
[15] Hill, Will, Larry Stead, Mark Rosenstein, and George Furnas. "Recommending and evaluating choices in a virtual community of use." In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 194-201. ACM Press/Addison-Wesley Publishing Co., 1995.
[16] R.Sugumar, A.Rengarajan, C.Jayakumar “Enhancement of K-Anonymity To Privacy Preserving Data Mining Using Association Rule Hiding Algorithm”, International Journal of Applied Engineering Research, Vol. 9, No. 24, pp. 26351-26364, November 2014.
[17] Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergström, and John Riedl. "GroupLens: an open architecture for collaborative filtering of netnews." In Proceedings of the 1994 ACM conference on Computer supported cooperative work, pp. 175-186. ACM, 1994.
[18] Resnick, Paul, and Hal R. Varian. "Recommender systems." Communications of the ACM 40, no. 3 (1997): 56- 58.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 7.542

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING