2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA)
Pimpri Chinchwad College of Engineering (PCCOE), Pune, India. Aug 26-27, 2022

2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA) | 978-1-6654-8451-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCUBEA54992.2022.10010973

AI in Entertainment – Movie Recommendation using cosine similarity

Karan Shah
Dept. of Computer Engineering Atharva College of Engineering (University of Mumbai)
Mumbai, India
karansshah00@gmail.com

Bhavna Arora
Dept. of Computer Engineering Atharva College of Engineering (University of Mumbai)
Mumbai, India
bhavnaarora@atharvacoe.ac.in

Aliraza Shinde
Dept. of Computer Engineering Atharva College of Engineering (University of Mumbai)
Mumbai, India
alishinde21@gmail.com

Shreyas Vaghasia
Dept. of Computer Engineering Atharva College of Engineering (University of Mumbai)
Mumbai, India
shreyasvaghasia01@gmail.com

***Abstract*— This age of information brings in a huge amount ofdata and combined with the number of options available on the internet, it becomes a huge problem in choosing a single thing. This is an example of the saying finding a needle in the haystack. The solution for that is using a magnet to attract the needle, the magnet in this case is a good recommendation system. A recommendation system is a tool that can help a user find the best option for the wide variety of options available. Nowadays every application we use is controlled by a recommendation system. Many platforms like Amazon Prime, Voot, ZEE5, which suggest movies, Zomato which suggests food, Spotify, Wynk, Hungama Music which suggests music, Airbnb, Trivago which suggests hotels and policy bazaar which recommends policy. In this project, we used Cosine similarity for a movie recommendation. Cosine similarity is a distance calculation metric that will tell us the distance between two items if the cosine of the angle between them is small it means that the items are close to each other and vice versa.**

Keywords—Cosine Similarity, Recommendation System, Personalization, OTT, Content-based filtering, Collaborative filtering

I. INTRODUCTION

In today's world, artificial intelligence is an integral part of the various industries as a whole. Especially during a pandemic where cinemas are closed and people working from home are doing something to calm down by watching movies or listening to music, these people are on the OTT platform. With the advent of Internet Service Providers, where on-demand streaming services offer almost unlimited content and deliver high-speed data at affordable prices the scenario has changed completely. These OTT services are catalogs of relevant content that provide endless topics for discussion. Now it has a big impact on our lifestyle. The impact of these services on our lives comes from subtle nuances, cultural references, and the cult of television shows and movies. Streaming services have evolved mainstream movies and televisions and brought their outfits to an ever-expanding user base.

II. RELATED WORK

There are various techniques of recommendation like Content-based, Collaborative based and hybrid models. Other recommendation systems use collaborative based which give recommendations purely based on ratings and genres. It does not take into consideration the user's experience and that is why we used collaborative filtering. A Hybrid System whichis a combination of both types of filtering can be used later aswe scale up the product. Different tiles on the personalizationpage can be given different recommendation techniques which in turn will improve the user experience. Collaborating is not a good idea when launching an application as it uses votes and ratings of other users. Mostly, an application will not have a tremendous user base in the beginning and hence collaborative filtering is not feasible.

Content-based approach has three methods of finding similarity namely :

A. Cosine Similarity:

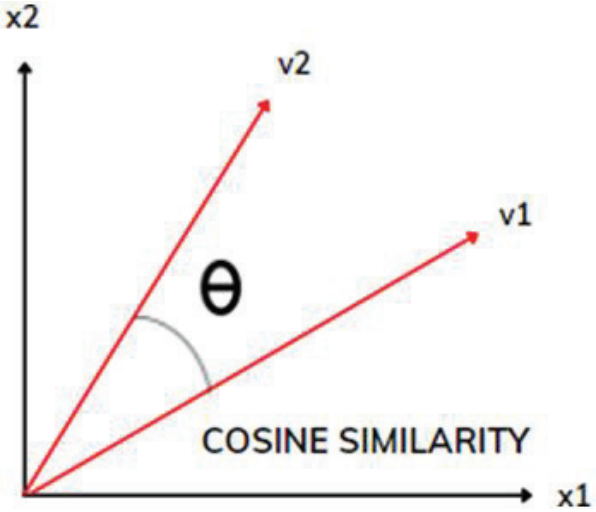

Fig. 1. Cosine Similarity

It is the Cos of the angle betweentwo vectors. The column's genres, keywords, cast and crew are concatenated and a new data frame "Tags" is created. Now, Tags are nothing but vectors. So, imagine a movie (tagone) is watched by a user. The recommendation will calculatethe nearest vector and provide a recommendation. So smallerthe angle, the higher the similarity between two vectors.

B. Euclidian Distance:

The euclidian distance is a tip to tip distance between two vectors. So, in a dataset like ours which has approximately 5000 movies, the results will not be accurate since euclidian distance fails in the high dimensional dataset. It is not a good and reliable measure in this case.

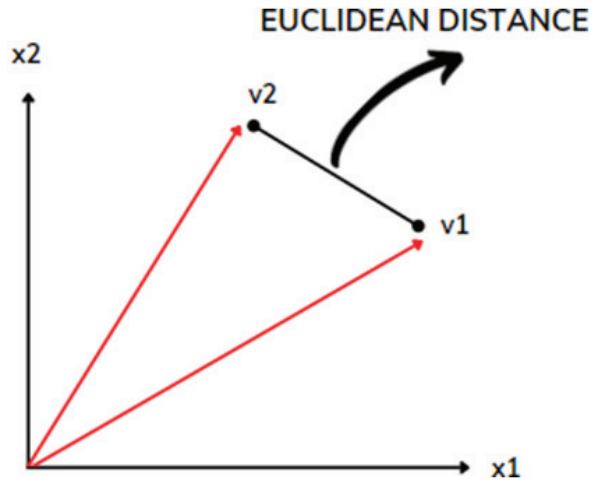

Fig. 2. Euclidean Distance

Pearson's Correlation - It computes the distance between two jointly distributed systems and hence a good algorithm for 2 vectors. But, in this project, we are dealing with around 5000 tags and hence it was not a good choice.

III. DATASET AND IMPLEMENTATION

Our project is mainly divided into five steps

1. Gathering the data.
2. Preprocessing the data
3. Building a model
4. Creating a website
5. Deploy and use it as a product

We used The Movie Database (TMDB) 5000 Movie Dataset. It has 2 files available. One was movies.csv which consisted of columns like budget, genres, homepage, id, keywords and much more.

Another was credits.csv which consisted of columns like title, cast, crew etc. This dataset was a good starting point, although there were other huge datasets available on Kaggle. According to our requirements and vision, TMDB 5000 was the best fit for us. We used both the files movies.csv and credits.csv which needed cleaning. We merged both datasets and included important columns from both of them. We discarded columns which played no role in our analysis. For example Budget, consider a movie that has a high budget and you liked the movie that does not mean that you will like another movie of high budget and hence we discarded it. Some of the important columns which we considered are genres, id, keywords, title, overview, cast and crew. We then created a unique data frame that consisted of movie-id, title and tags. We used a helper function to reuse computation just as the general function.

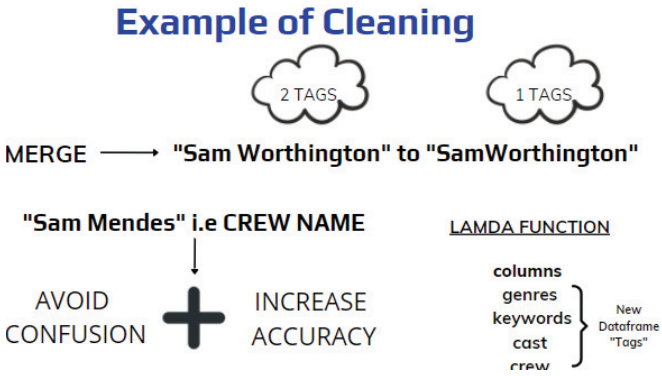

Fig. 3. Example of Cleaning

We had to convert lists of strings to lists for which we imported a module "ast" which has a function literal_eval. Another important thing was to merge the name and surnames of our cast and crew. For example, we had to merge "Sam Worthington" to "SamWorthington" to avoid confusion in our model because our model considers Sam as one tag and Worthington as another. For instance, we had a crew named "Sam Mendes" so there can be a possibility that we want movies of Sam Worthington but we get movies of Sam Mendes so we used the lambda function for columns genres, keywords, cast and crew to improve the accuracy. One of the difficulties was the textual data because it's not like numerical data in which we can use mathematical formulas and calculate the score that's where we needed vectorisation.

Fig. 4. Concept of Text Vectorisation

We converted texts to vectors which are known as text-vectorisation, we converted each movie tag into a vector so we can plot it in a 2D space and calculate the similarity. There are many techniques available for converting texts to vectors few of them are "Bag of words", "tfidf", "word2vec" and much more. We used a "Bag of words" which is one of the simplest techniques, so by this method we concatenated tags of all the movies which created a large text and from this large text available we found a frequency of all the words and found the top 5000 common words with the highest frequency and later extracted it by a class called "Count Vectoriser" in python.

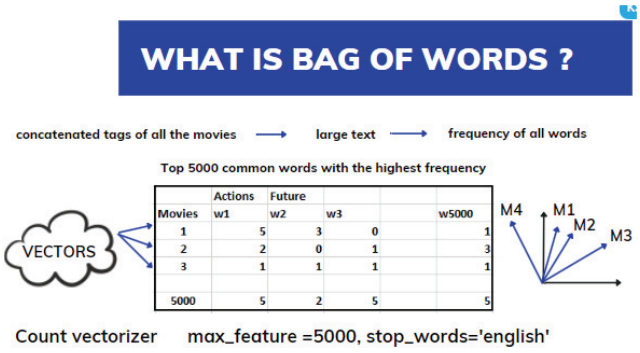

Fig. 5. Concept of Bag of Words

We used 2 main parameters i.e max_features and stop_words which eliminated common English words like are, to etc.

Now we received 5000 different words like act, acting, action, action-hero, actions, activist which are of similar types to solve this ambiguity and increase the accuracy we used stemming by which words for example act, acted, acting will be considered a single entity. For this, we used the NLTK library in python which is one of the famous libraries used in Natural Language processing.

IV. BASIC CONCEPT

Cosine similarity measures the similarity between two vectors of an inner product space. It determines whether two vectors are pointing in the same general direction by measuring the cosine of the angle between them. In text analysis, it's frequently used to determine document similarity.

Thousands of characteristics can be used to characterise a document, each of which records the frequency of a specific word (such as a keyword) or phrase in the document. Thus, each document is an object represented by what is called a *term-frequency vector*.

A metric is a unit of measurement that represents the distance between two items or how far apart they are. We can use another function called a similarity measure or similarity coefficient, or sometimes just a similarity, to assess closeness in terms of similarity.

We'll start with the set Rn and two vectors x,y∈ Rn. The dot product $x \cdot y$ is an operation on the vectors that returns a single number.

It is the equation x· y=∑i=1nxiyi.

The norm of a vector x is ‖ x‖ =∑i=1nxi2:

The **cosine similarity** is defined by the equation CS=x· y‖ x‖ ‖ y‖ .

V. DESIGN AND DEPLOYMENT

We have used react for the frontend along with the basics of Html, CSS and Javascript. Html is used to design basic structures and CSS to style web layouts. We used Mongo-Db, Express, Node for our backend. Mongo-Db to save the data, Express which is a flexible framework that provides a robust set of features for mobile and web applications. Our backend is deployed on Heroku and our frontend on Netlify.

We have used React for creating the front-end for our project.We use NodeJS,ExpressJs for the backend and MongoDB as out Database.The web-application is hosted on heroku server and the backend is hosted on the cloud . Also the mongoDb database is running on cloub ie Atlas server. We use Bootsrap for making our web -app responsive and mobile friendly.react-router-dom is used for routing and Redux is used for the state management for the entire application

The project code for the entire section is upload on the famous remote code sharing platform ie Github. we made two folder - client- for the frontend and server- for the backend. It can be downloaded on local system and can be run locally as well using the npm tool .

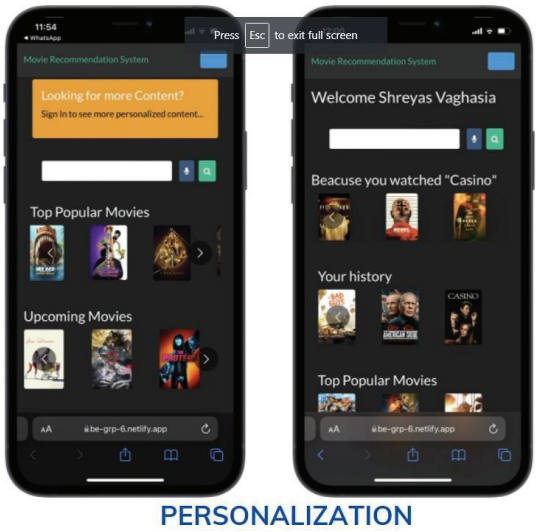

Fig. 6. Live Screenshot (Personalisation)

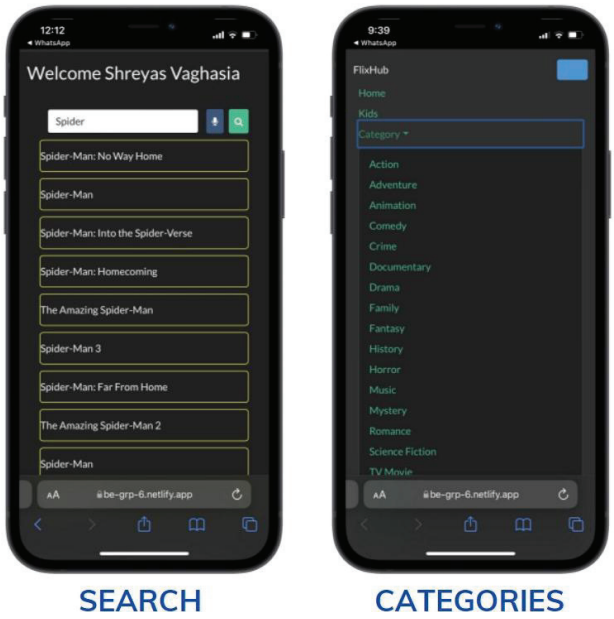

Fig. 7. Live Screenshot (Search & Categories)

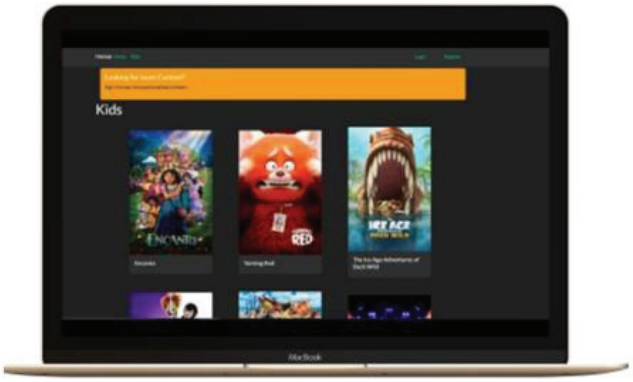

Fig. 8. Live Screenshot (Kidz Section)

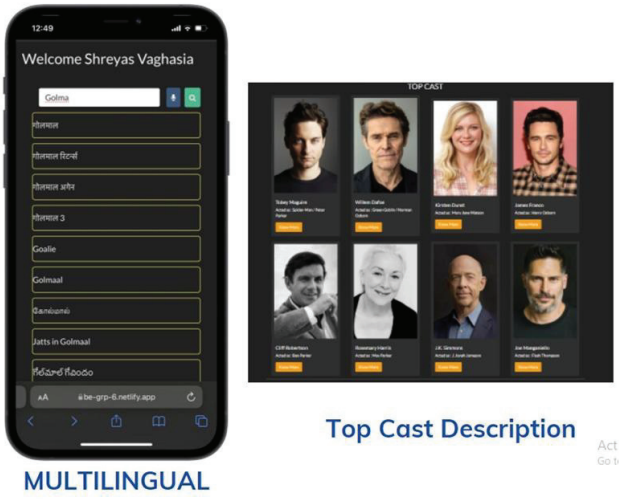

Fig. 9. Live Screenshot (Multilingual & Cast Description)

VI. FUTURE ENHANCEMENT

1. **Scaling** - When a project is used by millions of people home page along with the product must be strong enough to withstand the load. Also, auto-scaling is a feature that we would like to integrate in future.
2. **Investments -** There is a lot of money required for testing the product for instance 25 million people are streaming at the platform which would require C5 9X large machines to just generate the load. Each machine has 36 CPUs and 72 GB of RAM and 3000 such machines will be required to generate the load if it reaches such a level. We will need to move towards geo-distributed load generation because this high load of testing can disrupt other applications in the region.
3. **Chaos Engineering -** Here you know that failure is about to happen and how can team overcome it
4. **Performance and Tsunami Tests -** The sudden surge and dip are equally bad, you can still scale but backend servers, elastic cache are not scalable on the fly. This may be a possibility when a new movie is released.
5. **Different approaches -** We can move toward a hybrid approach and can personalize a recommendation in a specific tile.

VII. CONCLUSION

The dataset has been pre-processed using the pipeline mentioned in the Dataset and Implementation section of the paper. The original dataset is extracted from the TMDB API to train the model. The user uses the website and from the various search options can get the required results for the

REFERENCES

[1] Suresh, Salini, et al. “Latent Approach in Entertainment Industry Using Machine Learning.” International Research Journal on Advanced Science Hub, vol. 2, no. Special Issue ICARD, 2020, pp. 301–304., doi:10.47392/irjash.2020.106.

[2] Desai, Harshali. “Movie Recommendation System through Movie Poster Using Deep Learning Technique.” International Journal for Research in Applied Science and Engineering Technology, vol. 9, no. 4, 2021, pp. 1574–1581., https://doi.org/10.22214/ijraset.2021.33947.

[3] (PDF) AI in the Media and Creative Industries - Researchgate. www.researchgate.net/publication/333041972_AI_in_the_media_and _creative_industries.

[4] Sunilkumar, Chaurasia Neha. “A Review of Movie Recommendation System: Limitations, Survey and Challenges.” ELCVIA Electronic Letters on Computer Vision and Image Analysis, vol. 19, no. 3, 2020, p. 18., doi:10.5565/rev/elcvia.1232.

[5] Singh, Abhishek, et al. “A Research Paper on Machine Learning Based Movie Recommendation System.” International Research Journal of Engineering and Technology, vol. 08, no. 3, Mar. 2021, doi:10.26782/jmcms.2020.07.00056.

[6] Movie Recommendation System Using Cosine Similarity and KNN. www.researchgate.net/publication/344627182_Movie_Recommendation_System_using_Cosine_Similarity_and_KNN.

[7] “Cosine Similarity.” Wikipedia, Wikimedia Foundation, 11 Feb. 2022, en.wikipedia.org/wiki/Cosine_similarity.

[8] Lüthe, Marvin. “Calculate Similarity-the Most Relevant Metrics in a Nutshell.” Medium, Towards Data Science, 13 May 2021, https://towardsdatascience.com/calculate-similarity-the-most- relevant-metrics-in-a-nutshell-9a43564f533e. https://towardsdatascience.com/calculate-similarity-the-most- relevant-metrics-in-a-nutshell-9a43564f533e

[9] Lüthe, Marvin. “Calculate Similarity-the Most Relevant Metrics in a Nutshell.” Medium, Towards Data Science, 13 May 2021, https://towardsdatascience.com/calculate-similarity-the-most- relevant-metrics-in-a-nutshell-9a43564f533e.

[10] Liu, Jingdong, et al. “Personalized Movie Recommendation Method Based on Deep Learning.” Mathematical Problems in Engineering, Hindawi,19 Feb.2021, www.hindawi.com/journals/mpe/2021/6694237/.

[11] Movie Recommender System Using Critic Consensus. www.researchgate.net/publication/357268035_Movie_Recommender _System_using_critic_consensus/fulltext/61c3e8bc8bb20101842ed7dd/Movie-Recommender-System-using-critic-consensus.pdf.

[12] Content-Based Filtering Advantages & Disadvantages | Recommendation Systems | Google Developers.” Google, Google,developers.google.com/machine-learning/recommendation/content-based/summary.

[13] Shah, K., Shinde, A., Vaghasia, S. and Arora, B., 2022. AI IN ENTERTAINMENT - MOVIE RECOMMENDATION. [online] Irjet.net. Available at: <https://www.irjet.net/archives/V9/i2/IRJET-V9I2128.pdf> [Accessed 28 April 2022].