Proceedings of the 7th International Conference on Intelligent Computing and Control Systems (ICICCS-2023)
IEEE Xplore Part Number: CFP23K74-ART; ISBN: 979-8-3503-9725-3

Content-based Movie Recommendation System and Sentimental analysis using ML

Saketh Katkam[1], Abhishek Atikam[1], Pallerla Mahesh[1], Mrunal Chatre[1], Shree Sai Kumar[1], .Sakthidharan G R[2].,
[1] UG Student, Department of CSE, *Gokaraju Rangaraju Institute of Engineering & Technology,* Hyderabad, India
[2]Professor, Department of CSE, *Gokaraju Rangaraju Institute of Engineering & Technology,* Hyderabad
[1]saketh.katkam@gmail.com,

2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS) | 979-8-3503-9725-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICICCS56967.2023.10142424

***Abstract*—The recommendation system is important in today's world and is employed by many famous apps. A recommendations system is one that suggests books, movies, music, and other resources to users based on a data collection. Typically, movie recommendation algorithms anticipate what movies a user would enjoy based on the characteristics of previously loved movies. The algorithms can propose movies based on one or more criteria such as the genre of the film, the people in it, or even the director of the film. The method used is content-based filtering with genre correlation.**

Keywords—content-based movie recommender system, Application programming interface key, K-Nearest Neighbor algorithm, Natural Language Processing, Cosine Similarity, Sentimental analysis, Naïve Bayes classifier.

I. INTRODUCTION

A. Motivation

The world is so congested, recommendations are essential when promoting items or services. Data in such large volumes is sometimes inconsistent, and users must repeat their search several times before they ultimately find what they were looking for. To address this issue, the paperwork deals by using recommendation systems.

There are number of movies made every year as a result of the movie industry's recent growth and prosperity, making it challenging to locate a film that may be of interest to you. As a result, the issue of movie recommendations has gained popularity.

B. Related Work

Collaborative filtering and content-based filtering are the two broad categories into which recommender systems may be classified.

Using user-related data, preferences, and user-item interactions, collaborative filtering techniques can determine if two users are similar. It makes movie recommendations based on what other users who are similar to them prefer. Model-based and Memory-based methods can be used to further categorize them. Memory-based approaches don't require training since they compare how similar the "test" and "training" users are and then do a weighted average of those results to determine which suggestions to make.

The recommender systems employed by large corporations like Amazon, Netflix, etc. are common approaches in this category. Since at least ten years ago, Amazon has used collaborative filtering to suggest products to its customers. However, collaborative filtering techniques generally have a high processing cost and struggle with sparse data. Additionally, they make the unfounded assumption that consumers with comparable tastes will rate similarly.

Another common approach to addressing the suggestion issue is the content-based approach. Content-based approaches employ metadata from movies including narrative, review, cast, genres, and crew as its input, in contrast to collaborative filtering approaches (CF) which mostly uses user-item matrix input. Here, the user-recommended items are ones that are comparable to those they have accessed or searched.

Collaborative filtering and content-based recommender systems are fundamentally different from one another since the former relies solely on user ratings to generate suggestions, while the latter employs both user and item attributes to create predictions. Unlike CF approaches, content-based methods do not experience the cold start issue. Collaborative algorithms will start to perform worse when the number of users grows and the volume of data rises because of the sheer volume of data growth.

The method of clustering involves grouping a set of objects so that they are more like one another than to those in other clusters. On the movie dataset, the K-Nearest Neighbor clustering technique is used to get the best-optimized results.

II.METHODOLOGY

A. Dataset and Features

The IMDb Dataset (Information provided by IMDb at http://www.imdb.com) was used for our work. This dataset, which has over 6 million entries, includes information on all motion pictures, television programmers, documentaries, and short films that have been produced since 1910.The data comprises a range of movie-related details, including the title, ratings (which include user votes), runtime, crew (directors, writers, etc.), genre, and main cast (actors).

The proposed work utilized natural language processing to conduct sentimental analysis on the IMDB data. Web scraping the movie reviews from the IMDB website allows for sentimental analysis because it can categorize them as positive or negative based on the terms used in the dataset.

Authorized licensed use limited to: SRM Institute of Science and Technology. Downloaded on September 29,2023 at 04:53:41 UTC from IEEE Xplore. Restrictions apply.

The details of the movies, such as title, genre, runtime, rating, poster, casts, etc., are fetched from TMDB website (The Movie Data Base) using an API key. Here, The JavaScript file contains the API key. As a result, when a user searches for a movie, the TMDB website receives a request over this API and responds with the movie's information.

B. Architecture:

The suggested system's architecture is depicted in the form of a flow chart Figure-1. There are three separate components to it: an input module, a Front end, and a Backend. Figure-1 provides a conceptually sound explanation of how the suggested suggestion system operates.

The architecture of the system is then explained by a detailed depiction of each module. This makes it easier to grasp the system's design in a concise and understandable way.

Input module:

The user types the title of the movie into the website and clicks submit. The user merely needs to enter the title of the movie, and our website will return all relevant or suggested movie things.

Front End:

The website's front end comprises HTML and JavaScript code that enhances the user experience. The JavaScript file also contains the TMDB website API key, allowing the website to access the Movie information, including the title, poster, Genre, actors, and actor details, are retrieved from the TMDB website via API calls and from the IMDB website using web scraping.

Backend:

The Panda module in this first isolates the data from the raw files. Using the Panda's library, it divides the user and movie item data into distinct data frames. This helps in figuring out the subset of movie dataset that may be recommended to the user by its term frequency and inverse term frequency by its movie genre type. The meta data is converted into directed vectors using the vector space model. Each direction vector resembles a movie's characteristic, such as its genre or cast. KNN method is used to analyze the movie data with the aid of the vector space model, and a list of k movie sets that are like the desired movie is then obtained. These movie sets are once more subjected to cosine similarity processing, which ranks or arranges them in accordance with the degree to which they resemble the movie the user looked for.

Large volumes of data may be retrieved from websites via web scraping. By web scraping the movie reviews from the IMDB website, the paperwork uses the naive bayes algorithm to do sentimental analysis on the reviews. It is a feature of NLP (natural language processing) that separates positive from negative movie reviews in other terms good or bad reviews. When compared to reading through the entire review in detail, it is easier to completely comprehend or grasp a brief and simple notion.

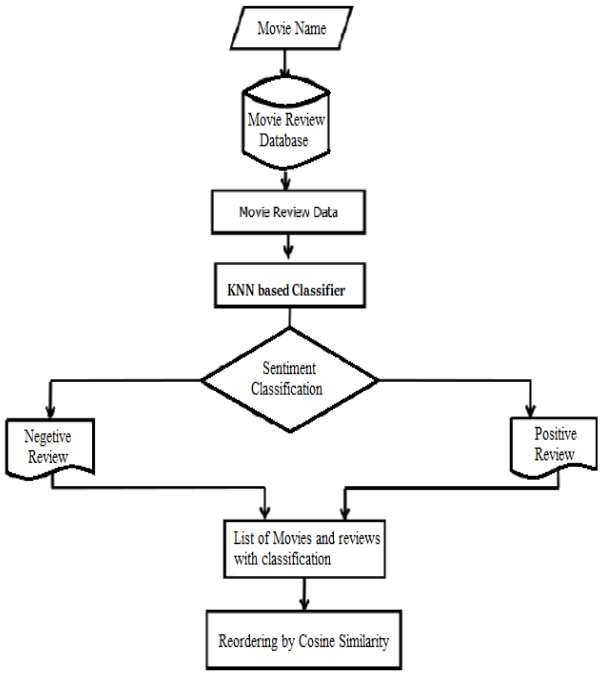

Fig1. Flow chart of Proposed System

The flask framework is used to show the data processed in the backend on the website. The website includes information on the user-searched movies, such as the movie's poster and cast, as well as reviews of the films that have undergone sentimental analysis. Finally, the website displays a list of suggested films based on the findings of the KNN and Cosine Similarity algorithms.

C. Algorithms:

The algorithms used in our proposed system were,

1.KNN (K Nearest Neighbors) algorithm:

A supervised machine learning approach for classification and regression issues is KNN, also known as K-nearest neighbors. One of the easiest algorithms to learn is K closest neighbors. Non-parametric, or K closest neighbors. For underlying data assumptions, it doesn't make any. As it keeps the data points rather than learning during the training phase, the K closest neighbors' method is also known as a lazy algorithm. The algorithm is based on distance.

The movie data must be in a vector field model for the KNN. In order to view the movie in terms of several properties like movie genre, director, actors, and other crucial elements, first insert the meta movie data into the vector space model. In order to get the k nearest datapoints—movies that are similar to the user-searched movie—the KNN method locates the closest points in the vector space model.

The output of KNN is the list of k movie dataset which are identical or similar to the required movie. The data is given input to the Cosine similarity algorithm to compute and reorders the list by the help of reviews and ratings of the movies.

Authorized licensed use limited to: SRM Institute of Science and Technology. Downloaded on September 29,2023 at 04:53:41 UTC from IEEE Xplore. Restrictions apply.

2. Cosine Similarity:

The goal of content-based recommender systems is to calculate the similarity between movies. Text, video, and images are all examples of content. Each movie is represented by a feature vector. The most often used metric of document similarity is cosine similarity.

To calculate the similarity of two Movies, us below Equation to compute the cosine of the angle between the features vectors.

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{n}^{i=1} A_i \times B_i}{\sqrt{\sum_{n}^{i=1}(A_i)^2} \times \sqrt{\sum_{n}^{i=1}(B_i)^2}}$$

Fig2. Formula for Cosine Similarity Calculation

i.e., A and B refers to the movies and θ(Theta) is the angle between them in Vector space.

The similarity ranges from -1 to 1. -1 indicates that the direction of the two vectors is diametrically opposed, and 1 indicates that they are pointing in the same direction.

If the two vectors have no relationship, the cosine similarity is 0. In order to match text, Because the weights are plainly nonnegative, the range must be 0 to 1 in our instance.

3. Naive Bayes Classifier:

A straightforward and effective classification job in machine learning is the Naive Bayes method. Naive Bayes classification is based on the Bayes theorem and a strong assumption of feature independence. The Naive Bayes classification performs well for textual data analysis, such as Natural Language Processing.

The Naive Bayes Classifier use the Bayes theorem to predict membership probabilities for each classification, such as the possibility that a certain document or data point belongs to that class. The class with the highest probability is referred to as the most likely class. This is also referred to as the Maximum A Posteriori (MAP).

csv files are loaded and converted into readable format using the pandas and NumPy packages. The web scraping of the IMDB website produced the csv files.

The review data is formatted properly by removing stopwords i.e., Stop words are phrases that are so frequent that standard tokenizers essentially disregard them. Next divide the data in part, 80:20 for training and test data. At last, perform the sentimental analysis by using Naive bayes algorithm in libraries like

```
from sklearn.model_selection import train_test_split
from sklearn import naive_bayes
from sklearn.metrics import
roc_auc_score,accuracy_score
```

The proposed model creates a random forest model on the training set and make predictions based on the features of the test set after splitting and vectorizing text evaluations into numbers.

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(x, y)
```

III.EXPERIMENT AND RESULT

The dataset with a sizable amount of movie information, or meta data, includes the name of the movie, the stars and producers, directors, as well as scores and reviews and information about the year and other factors obtained from multiple sources.

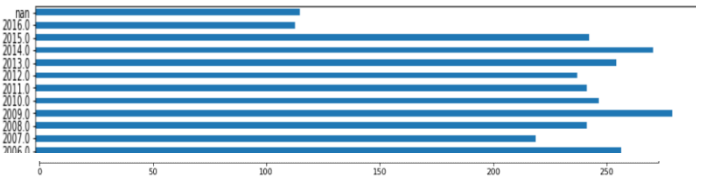

Fig 3. Horizontal Bar Chart of Movie Datasets by year.

The collection includes details on more than 6000 films (6010 rows X 7 columns) across various category categories. For the KNN Algorithm, this data is divided into 80% training and 20% testing after transforming the data with the aid of vector space model.

The Naive Bayes algorithm also divides the data into 80% training and 20% testing, allowing the evaluation to be categorized as either positive or negative with a random state of 42. The naïve Bayes accuracy scores are provided below.

```
clf = naive_bayes.MultinomialNB()
clf.fit(X_train,y_train)
```

```
MultinomialNB()
```

```
accuracy_score(y_test,clf.predict(X_test))*100
```

```
97.47109826589595
```

(i)

```
clf = naive_bayes.MultinomialNB()
clf.fit(X,y)
```

```
MultinomialNB()
```

```
accuracy_score(y_test,clf.predict(X_test))*100
```

```
98.77167630057804
```

(ii)

Fig 4. Accuracy Score for (i)Training data (ii)Dataset

Calculations are also done for the columns' sparsity. sparsity is the absence of data for a range of values in one or more dimensions. A matrix's sparsity is determined by dividing the number of cells that carry ratings by the maximum number of values that matrix may accommodate in context of the number of users and objects.

```
#Calculating the sparsity
no_of_users = len(ratings['userId'].unique())
no_of_movies = len(ratings['movieId'].unique())

sparsity = round(1.0 - len(ratings)/(1.0*(no_of_movies*no_of_users)),3)
print(sparsity)
```

```
0.99
```

Fig 5. Sparsity Calculation for columns ‘userId’ & ‘movieId’.

Authorized licensed use limited to: SRM Institute of Science and Technology. Downloaded on September 29,2023 at 04:53:41 UTC from IEEE Xplore. Restrictions apply.

IV. CONCLUSION

Analytical model construction is automated using machine learning, a technique for data analysis. It is a subset of artificial intelligence that is predicated on the notion that machines can learn from data, see patterns, and make judgements with little to no human involvement.

In this paper, content-based movie recommender system is proposed. The proposed method using KNN algorithm. The data are taken from IMDB and TMDB websites through API key. The system is implemented in python programming language. The recommender makes use of internet resources and assists in collecting relevant information so that it may be analyzed and used to produce appropriate suggestions depending on user requirements.

The system also performs the sentimental analysis on the web scraped reviews obtained from IMDB website. It enhances the efficiency of the system. The Naïve Bayes algorithm is used to classify the review into good or bad review. So, when compared to reading through the entire review in detail, it is easier to completely comprehend or grasp a brief and simple notion.

For future work, the recommendation system can use several ML methods, such as k-means and other algorithms, to quickly boost the recommender's efficiency. Increasing the data set also shows how effective the recommender is. hence a vast dataset should be supported by the system.

The recommendation systems used by services like Netflix and Amazon Prime are collaborative filtering techniques. It will also have its negatively affect i.e., cold start issue. Hence, by integrating with collaborative filtering approaches, content-based techniques must be developed so that it can manage both cold start and massive data challenges.

V. REFERENCES

[1] Rishabh Ahuja, Arun Solanki and Anand Nayyar, “Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor" 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 5386-5933-5/2019 IEEE.

[2] National Taiwan University of Science and Technology, “Fully Content based Movie recommendation system with Feature extraction using Neural networks”, 2017 International conference on Machine Learning, China 9-12 July,2017.

[3] Ruthann Singla, Saamarth Gupta, Anirudh Gupta, Dinesh Kumar Vishwakarma, “FLEX: A Content Based Movie Recommender”, 020 International Conference for Emerging Technology (INCET) Belgaum, India. Jun 5-7, 2020 ,7281-6221-8/2020 IEEE.

[4] Bagher Rahimpour Cami, Hamid Hassanpour and Hoda Mashayekhi, "A Content-based Movie Recommender System Based on Temporal User Preferences", 2017 3rd Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), 5386-4972/2017 IEEE.

[5] N. Pradeep, K. K. Rao Mangalore, B. Rajpal, N. Prasad and R. Shastri, “Content Based Movie Recommendation System”, International Journal of Research in Industrial Engineering, Vol 9

[6] H. Chen, Y. Wu, M. Hor and C. Tang, "Fully content-based movie recommender system with feature extraction using neural network," *2017 Int. Conf. on Machine Learning and Cybernetics (ICMLC)*, Ningbo, 2017, pp. 504-509.

[7] Charu C Aggarwal. An introduction to recommender systems. In *Recommender Systems*, pages 1–28. Springer,2016.

[8] K. S. Jones, “A Statistical Interpretation of term specificity and its application in retrieval”, *Journal of Documentation,* (vol. 60 no. 5), 2004, pp. 493-502

[9] S. E. Robertson,” Understanding inverse document frequency: on theoretical arguments for IDF”, *Journal of Documentation*, (vol. 60 no. 5), 2004, pp 503–520.

[10] O. Shahmirzadi, A. Lugowski and K. Younge, “Text Similarity in Vector Space Models: A Comparative Study”, arXiv:1810.00664, 2018, pp. 659-666.

Authorized licensed use limited to: SRM Institute of Science and Technology. Downloaded on September 29,2023 at 04:53:41 UTC from IEEE Xplore. Restrictions apply.