**Movie Recommendation System based on User Preferences**

**ABSTRACT: -**

A Recommendation System is a filtering program whose primary goal is to predict the “rating” or “preference” of a user towards a domain-specific element. In our project, this domain-specific element is a movie. Hence the main focus of our recommendation system is to provide a total of ten movie recommendations to users who searched for a movie that they like. These results are based on similar traits/demographics of the movie that has been searched. Content based filtering is a technique that is used to recommend movies. Apart from providing recommendations the system also provides posters of the Movies along with Release Date, Budget, Revenue, Popularity, Similarity between selected Movie and an Overview as well. The System uses the concept of vectorization based on common features and uses Cosine Similarity with respect to each other vectors to determine the most similar movies.

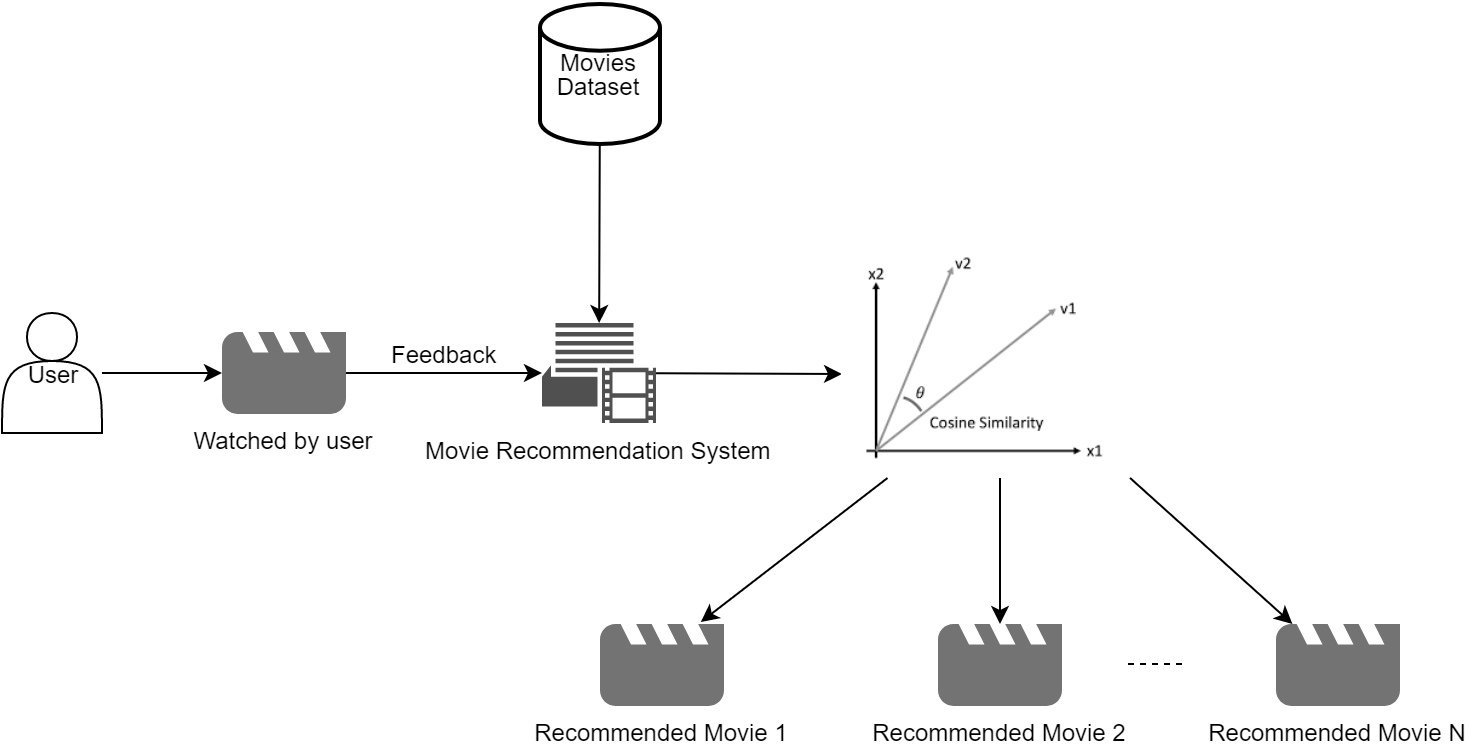
Batch: A8

Details:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Hall Ticket No.** | **Name** | **Contact** | **Email Id** |
| **1.** | 19WJ1A0519 | Anagondi Shiva Prasad | 9347025164 | shiva.prasad3345@gmail.com |
| **2.** | 19WJ1A0531 | Badempet Pavan | 7989256101 | pavan9b@gmail.com |
| **3.** | 19WJ1A0561 | Cheerala Prashanth | 9347604784 | cheeralaprashanth10@gmail.com |

|  |  |
| --- | --- |
| **EXISTING SYSTEM** | **PROPOSED SYSTEM** |
| * Recommender systems are information ﬁltering tools that aspire to predict the rating for users and elements, predominantly from big data to recommend their likes. Movie recommendation systems provide a mechanism to assist users in classifying users with similar interests. This makes recommender systems essentially a central part of websites and e-commerce applications. * This article focuses on the movie recommendation systems whose primary objective is to suggest a recommender system through data clustering and computational intelligence. In this research article, a novel recommender system has been discussed which makes use of k-means clustering by adopting cuckoo search optimization algorithm applied on the Movies dataset. | * The main purpose to develop a movies recommendation system is to provide users with recommendations that are not based on popularity or purely rating but based on the movies that the user likes. This will lead to a highly personalized recommendation, which will increase the accuracy of the recommendation system. * Providing Posters for Movies also helps the user won’t have to surf the internet for finding a movie that he/she likes. The user won’t have to rely on friends for a movie suggestion as the recommendation system will provide the user with the top ten movies that are most similar to the searched movie. |
| **EXISTING ALGORITHM**   * k-means & cuckoo search optimization | **PROPOSED ALGORITHM: -**   * Cosine Similarity Technique |
| **ALGORITHM DESCRIPTION**   * We use k-means as clustering algorithm and cuckoo search as optimization algorithm and then apply to Movies dataset for improved efﬁcient recommender systems Initially k-means clustering algorithm is applied to Movies dataset for clustering of users into different clusters. * The clusters are selected randomly at ﬁrst then users are inspected one by one by calculating the differences in their ratings and the centroid of the clusters, and if their difference is smallest, then the user gets allocated to the cluster to which they are closest. However, at this moment does not assure that each user has been assigned to the real cluster with a minimum difference of centroid. So, each user’s distance is compared to its cluster mean and with other clusters mean and relocate the users according to the smallest distance from any cluster’s mean. Next cuckoo search optimization algorithm is applied to the resultant of the k-means algorithm for optimizing the results. | **ALGORITHM DESCRIPTION**   * The movies are recommended based on a simple algorithm called Cosine Similarity. Cosine similarity is a measure used to determine the similarity between two elements. Mathematically it can be determined as the cosine angle between two vectors in a three-dimensional plane. We can also check the Euclidean distance between the two vectors to determine how different or similar they are from each other. In our case, one of the vectors is the movie that is searched and the rest of the movies in the database are checked as the second vector. The top ten movies which have the least Euclidean distance corresponding to the searched movie are shown as recommendations. * Cosine Similarity is a type of Content-based filtering approach. It is one of the most popular techniques used in recommendation systems. The attributes of a thing are termed as “content”. Based on these attributes we can classify whether the two things are similar or not. The attributes can be words specified in the database such as genre, cast names, director names, description, and so on. If the attributes match or have a high similarity, then the two movies can be classified as similar movies. The intuition behind this sort of recommendation system is that if a user liked a particular movie or show, he/she might like a movie or a show similar to it. |
| **DRAWBACKS**   * The slow rate of convergence. * It has trouble clustering data where clusters are of varying sizes and density. | **ADVANTAGES**   * Cosine is not affected by vector length. Thus, switching to cosine from dot product reduces the similarity for popular content. * Measuring the similarity between two vectors even if the data duplication is there. * It easily falls into the local optimal solution. |

**SYSTEM ARCHITECTURE**

****

**Fig. Proposed Methodology**

**MINIMUM SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| PROCESSOR | : | Pentium Processor |
| RAM | : | 2GB DDR3 RAM |
| HARD DISK | : | 5 GB Free Disk Space |

**SOFTWARE REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| FRONT END | : | Streamlit Python |
| BACK END | : | Azure, Docker, Python |
| OPERATING SYSTEM | : | Windows 7/8/10 |
| IDE | : | Visual Studio Code |