TV Series Recommendation Using Fuzzy Inference System, K-Means Clustering and Adaptive Neuro Fuzzy Inference System

Muyeed Ahmed, Anirudha Paul, Mir Tahsin Imtiaz, Md. Zahid Hassan, Shawon Ashraf, Rashedur M Rahman
Department of Electrical and Computer Engineering, North South University
Plot-15, Block-B, Bashundhara, Dhaka, Bangladesh
akib100095@gmail.com, anirudhaprasun@gmail.com, tahsin.imtiaz@northsouth.edu, mdzahidh119@gmail.com, shawon.ashraf@northsouth.edu, rashedur.rahman@northsouth.edu

Abstract—**Recommending TV Series is a more challenging task than movie recommendation. Not only the system should consider the taste of the user, it has to take into account the time commitment factor because a TV series can contain thousands of episodes. This paper proposes a way of recommending TV series by analyzing the users' genre preferability of movies, the genre of the TV series and the number of episodes. This system analyzes the genre preferability of the user from movie data using Fuzzy Inference System, puts the users of similar taste into a cluster using K-Means and finally applies Adaptive Fuzzy Neuro Inference System in the cluster to predict the rating of that TV series the user might give in real life.**

Keywords- ANFIS, Collaborative Filtering, Fuzzy Logic, TV Series Recommender System, Neural Network.

I. INTRODUCTION

There are many TV series streaming services like - Netflix [1], Hulu [2], Amazon Video [3] etc. and they try to recommend movies and TV series based on user's taste or general popularity among their users. But there is always a room for improvement. In this paper, we have derived a different way of recommendation. This system recommends TV series based on the movies a user likes. The main reason behind this choice is that, in a lifetime a user generally watches a vast number of movies compared to TV series. So, movies are better choice for analyzing the content consumption preferability of the individuals. In this paper, movie data are collected from MovieLens [4] and TV series data from IMDB [5].

The objective of this paper is to predict what rating a user might give to a certain TV series by analyzing information about the user and TV series. If the rating is within an acceptable range for the user, we will suggest the TV series to that individual.

In this recommendation process, we first take our collected MovieLens [4] data to find out the average rating and average ratio of each genre for a particular user. From this information, the system derives the fuzzy preferability of users for each of the 18 genres of the movies. By analyzing the genre preferability, the users are then divided into clusters by using K - means algorithm [6]. These clusters represent the groups who have comparatively similar taste in terms of watching movies. Then we take the new users from IMDB [5] and analyze their movie data and assign each of them to a cluster that is the closest to them by using Manhattan Distance. After assigning every user to any of the pre-defined cluster of similar taste, an Adaptive Fuzzy Neuro Inference System (ANFIS) [7] structure is built for every cluster. To train the ANFIS, we have used the TV series ratings given by the users in the cluster along with some other related variables like genre preferability and overall desirability of the TV series. For testing, we have used the data of new users in the system. Then we compare the system output with the original rating given by users to check the performance of our system.

II. RELATED WORK

The authors in [8] proposed a recommendation mechanism which was based on properties of the users. The data for the system includes user activity, interests of users, moods and experiences of the users and users' demographic information which is the input for the neural network they designed. This network predicts TV program preferences of the user. Pigeau, et al. [9] have presented a system that used a summarization method to recommend TV series. This technique enables the system to learn automatically by scanning the user profiles. The architecture is based on SAINTETIQ model which builds a summary tree by applying conceptual clustering. This enables structured data classification that is stored into the database. The next paper [10] has proposed an expert movie recommendation system that is implemented using machine learning. It also uses cluster analysis. A SVM is used to predict movies by analyzing user information such as age, gender, occupation, area and hobby. In the system, movies are selected and questions are prepared for the users. From the answer given by the user, the movie set is refined and the movies get finally recommended to the user.

Siddiquee, Haider and Rahman [11] analyzed the improvement in recommendation using FIS and ANFIS. The recommendation is done by analyzing choices of similar users and analyzing which genres are rated by the user. The next paper [12] explains the importance of building a recommendation system that considers how many special features a product has. They have proposed a system that analyzes the needs of the user and retrieve optimal products in order to recommend them to the user. Fuzzy logic and data mining is used in the system. The system helps to prepare information about the products that has high potential for

978-1-5386-2165-3/17/$31.00 ©2017 IEEE

being recommended to certain user. Velusam et al.[13] have proposed a single end-to-end advertisement recommendation system that analyzes content of ad/program, interests of the viewers', preferences that sponsors' have, program timing, popularity of the program and available slots for advertisement to recommend a set of advertisements that is scheduled well and sequenced and also suited best for a certain TV advertisement break.

III. SYSTEM DESIGN

In order to recommend a TV series, the initial step of our recommendation system is to divide our whole user database in clusters so that each of the clusters contains users who have relatively similar taste in terms of media consumption. For achieving this goal, our recommendation system initially uses MovieLens [4] users' movie data to create 10 clusters and also calculates their center for later part of our analysis. At first, we have made a fuzzy inference system which takes average rating and consumption ratio for each genre as input and generates genre preferability per user. This data indicates the fondness of a user to a particular genre. After calculating the genre preferability, we have used k-means clustering to divide all the users in 10 separate clusters and calculated their center.

After that, we take IMDB [5] user data. In order to assign them to the previously calculated clusters, we have applied the same fuzzy inference system to determine their genre preferability. We have calculated Manhattan distance to

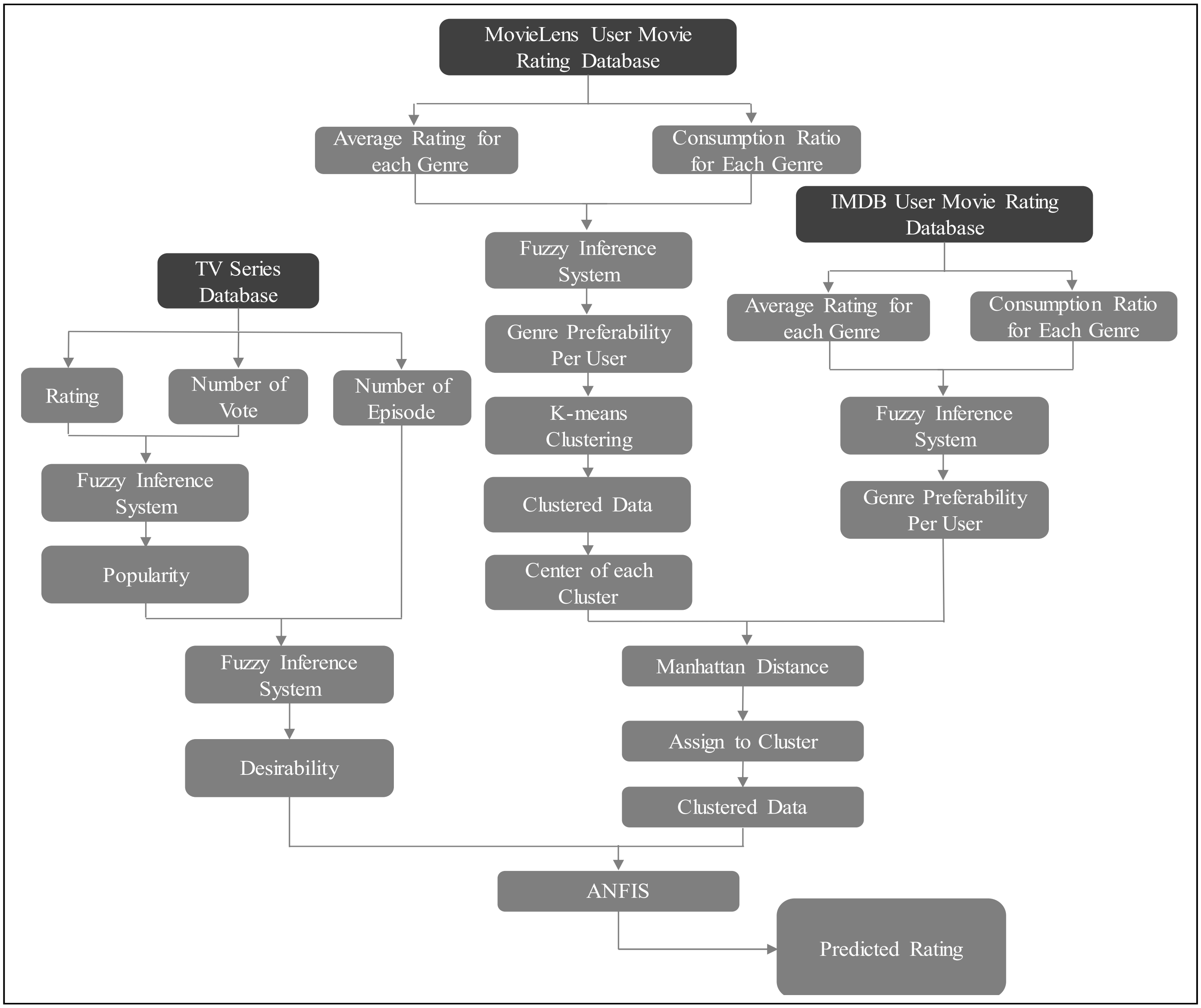

Figure 1. System Architecture

each of the center of the 10 clusters from all new users and found the closest cluster for each user. Finally, in order to recommend a TV series, this system analyzes all the user's TV series ratings. For every user, this analysis is done only within the users of the same cluster. This whole analysis and prediction is done using ANFIS system we have built, so that every cluster contains its own ANFIS structure. The system architecture of this system is shown in Fig. 1.

A. User Preferability Using Fuzzy Inference System

The objective of this step is to find the preferability of individuals for every genre of movies as we want to use this information to make cluster of similar type of users. For this part, we used MovieLens [4] data as it contains many movie data and user ratings. This Fuzzy Inference System takes two sets of data as input. The first set of input is the average rating a user has given to a particular genre as it shows how much the user likes movies from that particular genre. The second set of data represents the average media consumption in that particular genre. Because the more movies of a genre a user watches, the more preferable it is to that user.

To calculate the average rating, it adds all the rating the user gave to movies and divides it by total number of movies watched and rated by the user. To calculate ratio, it divides the number of movies of a genre a user rated by total number of movies that the user watched and rated.

For example, using data listed at TABLE I and TABLE II, the system calculates average rating and ratio shown in TABLE III and TABLE IV respectively for comedy movies.

TABLE I. RATING GIVEN BY USERS TO MOVIES

	Toy Story	Braveheart	Rush Hour
User1	7	10	-
User2	8	9	7
User3	9	-	9

TABLE II. GENRE OF MOVIES

	Genre 1	Genre 2	Genre 3
Toy Story	Animation	Adventure	Comedy
Braveheart	Biography	Drama	History
Rush Hour	Action	Comedy	Crime

TABLE III. CALCULATING AVERAGE RATING FOR COMEDY MOVIES

	Total Rating	Number of Comedy Movies Watched	Average Rating
User1	7	1	7
User2	15	2	7.5
User3	18	2	9

TABLE IV. CALCULATING RATIO FOR COMEDY MOVIES

	Number of Comedy Movies Watched	Total Number of Movies Watched	Ratio
User1	1	2	0.5
User2	2	3	0.66
User3	2	2	1

TABLE V shows membership function of both rating and ratio, and TABLE VI shows the membership function of the output, preferability.

1) Input Variables

a) Rating: Gaussian Curve is used to input the Rating variable. The input range is [0 10] and we defined this range into 5 linguistic membership function (Table V).

b) Ratio: Triangular Curve is used to input the Ratio variable. This input range is [0 1] and this range is also divided into 5 linguistic membership function (Table V).

TABLE V. MEMBERSHIP FUNCTION OF THE INPUT VARIABLES OF PREFERABILITY

Input	**Type**	**Parameters**	**Classification**
Rating	Gaussian	[0.5 0]	Very_Low
		[0.5 1.25]	Low
		[0.5 2.5]	Medium
		[0.5 3.75]	High
		[0.5 5]	Very_High
Ratio	Triangular	[0 0 0.25]	Very_Low
		[0 0.25 0.5]	Low
		[0.25 0.5 0.75]	Medium
		[0.5 0.75 1]	High
		[0.75 1 1]	Very_High

We used triangular membership function for the input ratio as it's simpler and have less variations in it than rating.

2) Output Variable(Preferability)

Gaussian Curve is used for the output curve to achieve maximum smoothness as our output does not contain any asymmetric membership function. The output range is [0 1] and the table is given in Table VI.

TABLE VI. MEMBERSHIP FUNCTION OF THE OUTPUT VARIABLES OF PREFERABILITY

Output	**Type**	**Parameters**	**Classification**
Preferability	Gaussian	[0.125 0]	Hate
		[0.125 0.25]	Dislike
		[0.125 0.5]	Mediocre
		[0.125 0.75]	Like
		[0.125 1]	Favorite

TABLE VII shows the rule by which this output was generated and Fig. 2 shows the surface plot of user preferability.

TABLE VII. USER PREFERABILITY RULES

		Ratio				
		Very Low	**Low**	**Medium**	**High**	**Very High**
Rating	**Very Low**	Hate	Hate	Dislike	Dislike	Mediocre
	Low	Hate	Dislike	Dislike	Mediocre	Like
	Medium	Dislike	Dislike	Mediocre	Like	Like
	High	Dislike	Mediocre	Like	Like	Favorite
	Very High	Mediocre	Like	Like	Favorite	Favorite

In TABLE VIII, we have shown the prefarability from average rating and ratio for comedy movies from the input data we calculated from TABLE III and TABLE IV.

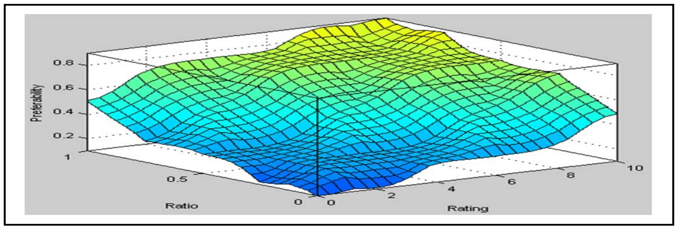

Figure 2. Surface of User Preferability

TABLE VIII. PREFERABILITY EXAMPLE FOR COMEDY MOVIES

Comedy Movies			
	Rating	**Ratio**	**Preferability**
User 1	7.5	0.66	0.723
User 2	7	0.5	0.7
User 3	9	1	0.89

B. Divide Users Into Clusters Using K-Means clustering

In order to divide the users into groups of similar tastes, we have calculated 10 clusters. Each of the clusters represents users of similar media consumption taste. The plotting variables for clustering are the preferability of the users for each of the genre we have calculated in the previous step. So, it is an 18-dimensional plotting as there are 18 genre preferability variable for every user. Clustering is used to divide the user data set into 10 different groups. This clustering has been done with simple K- means algorithm. For example, the center of a cluster is shown in TABLE IX.

TABLE IX. CLUSETER CENTER EXAMPLE

Action	Adventure	Animation	Children	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi	Thriller	War	Western
0.37	0.48	0.55	0.23	0.49	0.55	0.09	0.45	0.12	0.16	0.23	0.11	0.46	0.42	0.34	0.49	0.20	0.05

K means is used over Fuzzy C means because each of the user needs to be belong to only one cluster set for ANFIS analysis and it runs faster than the Fuzzy C means [6].

C. Assigning IMDB Users' to Cluster

In order to analyze TV series related data in our system, we have collected IMDB [5] data as it has both tv series and movie data. In order to assign them to our previously built clusters, we went through almost similar procedure for this data set. The difference is that this IMDB data contains TV series ratings along with movie ratings.

1) Calculate Preferability: This system uses the same fuzzy inference system as explained before to calculate prefarability of each IMDB [5] user for all genre.

2) Assignning to Cluster: The system uses Manhattan distance to assign each user to a cluster. It calculates Manhattan distance from a tuple to all the cluster centers and it sends the tuple to the cluster with minimum distance from that tuple. For simplicity, we can consider the example that is shown in the following table X. There is one user who has the preferability for action and adventure genre. There are two clusters having the same property as cluster centers. For example, TABLE X shows preferability for two users for two genres - action and adventure. To find preferable cluster TABLE XI shows two cluster center, c1 and c2 of two clusters, cluster 1 and 2.

TABLE X. PREFERABILITY OF TWO USER FOR TWO GENRE

	Preferability	
	Action	Adventure
User 1	0.7	0.4
User 2	0.55	0.6

TABLE XI. PREFERABILITY CENTER OF TWO CLUSTER FOR TWO GENRE

	Preferability	
	Action	Adventure
c1	0.6	0.65
c2	0.8	0.5

Manhattan distances from the user data to the cluster centers are calculated and the user is assigned to the cluster that has the minimum distance from the user data. For this example, for user 1, the distances from c1 and c2 are 0.35 and 0.2 respectively and for user 2, the distance from c1 and c2 are 1.0 and 0.35 respectively. Therefore, user 1 and user 2 belong to cluster 2 and cluster 1 respectively.

D. TV Series Desirability Using Fuzzy Inference System

In this part, the system will use fuzzy inference system to determine desirability of each TV series currently in the database. As desirability is a fuzzy output obtained from considering the rating, number of votes and number of episodes of that TV series and these values play a vital role for a user while selecting a new TV series. It will consider IMDB [5] rating, number of votes and number of episodes. Initially it will calculate popularity using rating and number of votes as we think not only rating but also number of votes indicates how popular a TV series is. After that it will calculate desirability using popularity and the number of episodes. Because we believe that the number of episodes is an important factor for a TV series to be desirable. A series with few numbers of episodes is more desirable than a series with many numbers of episodes as the user does not need a lot of time to finish that series. TABLE XII shows the format of how the dataset is stored in the database.

TABLE XII. TV SERIES DATA

Name	Rating	Number of Votes	Number of Episode
Game of Thrones	9.5	1093772	73
South Park	8.8	249425	281
Planet Earth II	9.8	13151	6
Family Feud	7	1631	407

1) Popularity Calculate: The system will calculate popularity using fuzzy inferance system taking IMDB [5] rating and number of voters as input.

a) Input Variables: Input variables are rating and number of votes.

i) *Rating:* Sigmoidal curve was used for characterizing the membership function of VeryLow and VeryHigh and gaussian curve for Low, Medium and high. The input range is [0 10]. Membership function plot of input variable TV series rating is shown in Fig. 3.

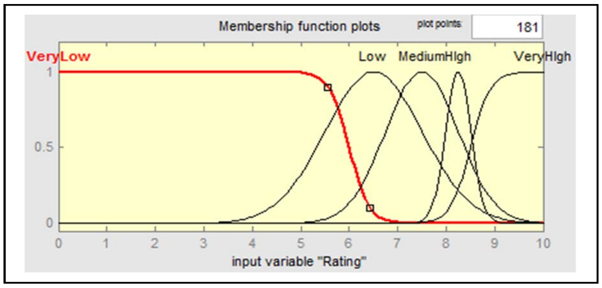

Figure 3. Membership Function Plots of Input Rating

ii) *Number of Votes:* Gausian curve was used for characterizing the membership functions VeryLow and Low. For membership functions Medium and High gaussian combination curve was used and for VeryHigh sigmoidal curve was used. The input range is [0 1200000]. Membership function plot of input variable number of vote is shown in Fig. 4.

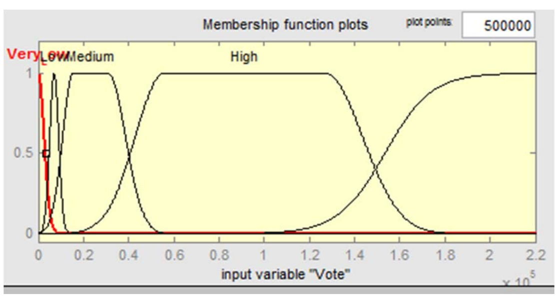

Figure 4. Membership Function Plots of Input Vote

TABLE XIII. MEMBERSHIP FUNCTION OF THE INPUT VARIABLES OF POPULARITY

Input	Type	Parameters	Classification
Rating	Sigmoidal	[-5 6]	Very_Low
	Gaussian	[1 6.5]	Low
	Gaussian	[0.75 7.5]	Medium
	Gaussian	[0.25 8.25]	High
	Sigmoidal	[5 8.5]	Very_High
Vote	Gaussian	[2600 200]	Very_Low
	Gaussian	[2000 7000]	Low
	Gaussian combination	[4700 15500 7700 30900]	Medium
	Gaussian combination	[13000 56000 16500 130000]	High
	Sigmoidal	[0.000097 153400]	Very_High

After studying our data set, we designed the membership functions described in TABLE XIII as it well distributes the ratings and votes from the user.

b) Output Variable(Popularity): Gaussian Curve is used for the output curve. The output range is [0 10]. Membership function plot of output variable popularity is shown in Fig. 8.

TABLE XIV. MEMBERSHIP FUNCTION OF THE OUTPUT VARIABLES OF POPULARITY

Output	Type	Parameters	Classification
Popularity	Gaussian	[0.125 0]	VeryUnpopular
		[0.125 0.25]	Unpopular
		[0.125 0.5]	ModeratelyPopular
		[0.125 0.75]	Popular
		[0.125 1]	VeryPopular

TABLE XV shows the rules on how the output was generated. Fig. 5 shows the surface of TV series popularity.

TABLE XV. TV SERIES POPULARITY RULES

		Vote				
		Very Low	**Low**	**Medium**	**High**	**VeryHigh**
Rating	**Very Low**	Very Unpopular	Very Unpopular	Very Unpopular	Unpopular	Moderately Popular
	Low	Very Unpopular	Unpopular	Unpopular	Unpopular	Moderately Popular
	Medium	Unpopular	Unpopular	Moderately Popular	Moderately Popular	Moderately Popular
	High	Unpopular	Moderately Popular	Popular	Popular	Very Popular
	Very High	Moderately Popular	Popular	Popular	Very Popular	Very Popular

In table XVI we can see that Game of Thrones has the highest popularity because it has both very high rating and number of voters is extremely high, On the other hand Planet Earth 2 has higher rating but it falls behind the popularity of Game of Thrones due to lack of voters. On the other hand, South Park has lower rating but it has really high number of voters so its popularity is also higher than Planet Earth. The last data Family Feud has extremely low popularity due to poor rating and poor number of votes.

TABLE XVI. POPULARITY EXAMPLE

	Rating	Number of Vote	Popularity
Game of Thrones	9.5	1093772	7.63
South Park	8.8	249425	7.1
Planet Earth II	9.8	13151	6.69
Family Feud	7	1631	2.27

After calculating popularity for all the TV series from our database, we saw that Braking Bad, Game of Thrones, The Wire, The Soprano and Friends were among top 10 popular movies having a popularity score of 7.64, 7.63, 7.59, 7.47, 7.33 respectively. We tested them against real world data from other sources for validation. Our no 1 spot is taken by Breaking

Bad which broke the world record for highest rated TV series [14]. Our second spot is taken by Game of Throne which has a world record of winning most Emmy Award in fictional series category and also winning most Emmy Award in drama series category [15]. The Soprano has captured the top position in the list of 101 Best Written TV series given by the Writers Guild of America [16]. Friends is at the 4th position in the list of Fig.5.

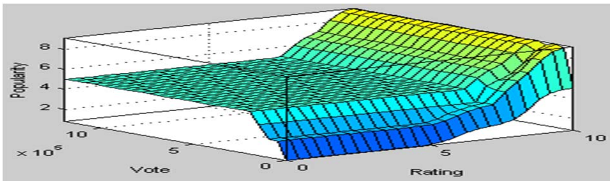

Figure 5. Surface plot of TV series popularity

Most Watched TV Series Finale with a massive viewership of 65.9 Million views [17].

2) Desirability Calculation: In this part the system will use fuzzy inference system to calculate disirerability. It will take previously calculated popularity and number of episodes as input and calculate desirability for each TV series in the database. Desirability creates a bridge between the drive for watching the TV series due to its popularity and the amount of dedication needed towards the series in order to complete the whole series.

a) Input Variables: Input variables are popularity and number of episodes.

i) *Popularity:* Gaussian Curve is used for the input curve. The input range is [0 10]. Membership function plot of input variable popularity is shown in Fig. 10.

ii) *Number of Episodes:* Gaussian Curve was used for characterizing the membership function of Very_Low, Low and Medium. Gaussian combination curve uas used for High and sigmoidal curve was used for Very_High. The input range is [0 1000]. We observed from our data set that there are few number of series with many number of episodes. Most of the series contains less than 200 episodes. Therefore, to properly distribute the TV series we designed the membership functions described

iii) in TABLE XVII. Membership function plot of input variable number of episode is shown in Fig. 6.

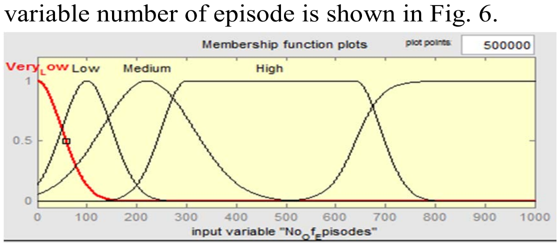

Figure 6. Membership Function Plots of Input Number of Episode

b) Output Variable(Desirability): Triangular Curve is used for the output curve desirability. The output range is [0 1].

TABLE XVII. MEMBERSHIP FUNCTION OF THE INPUT VARIABLES OF DESIRABILITY

Input	Type	Parameters	Classification
Popularity	Gaussian	[0.125 0]	VeryUnpopular
		[0.125 0.25]	Unpopular
		[0.125 0.5]	ModeratelyPopular
		[0.125 0.75]	Popular
		[0.125 1]	VeryPopular
Number of Episodes	Gaussian	[50 0]	Very_Low
	Gaussian	[50 100]	Low
	Gaussian	[100 220]	Medium
	Gaussian combination	[50 320 50 600]	High
	Sigmoidal	[0.03 640]	Very_High

TABLE XVIII. MEMBERSHIP FUNCTION OF THE OUTPUT VARIABLES OF DESIRABILITY

Output	Type	Parameters	Classification
Desirability	Triangular	[0 0 0.25]	Very_Low
		[0 0.25 0.5]	Low
		[0.25 0.5 0.75]	Medium
		[0.5 0.75 1]	High
		[0.75 1 1]	Very_High

TABLE XIX shows the rules on how the output desirability was generated. Fig. 7 shows the surface of TV series desirability.

TABLE XIX. TV SERIES DESIRABILITY RULES

		Number of Episode				
		Very Low	**Low**	**Medium**	**High**	**Very High**
Popularity	**Very Unpopular**	Low	Low	VeryLow	VeryLow	VeryLow
	Unpopular	Low	Low	Low	VeryLow	VeryLow
	Moderately Popular	Medium	Medium	Medium	Low	Low
	Popular	High	High	High	Medium	Medium
	Very Popular	VeryHigh	VeryHigh	VeryHigh	High	High

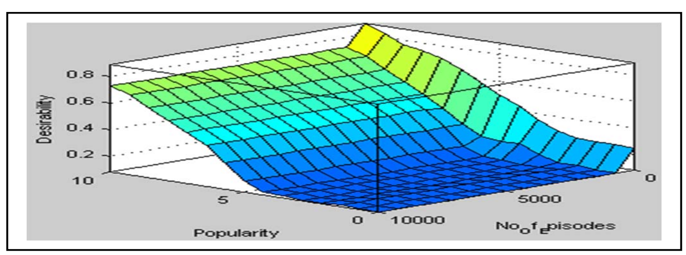

Figure 7. Surface of TV Series Desirability

In table XX we can see that due to having only 6 episodes the desirability of Planet Earth 2 is higher than South Park though it is less popular. Minimal effort to complete the series ranked it higher. On the other hand, Game of Thrones despite having significantly more episode numbers than the Planet Earth II ranked higher in terms of desirability due to its extreme popularity.

TABLE XX. DESIRABILITY EXAMPLE

	Popularity	Number of Episode	Desirability
Game of Thrones	7.63	73	0.73
South Park	7.1	281	0.61
Planet Earth II	6.69	6	0.67
Family Feud	2.27	407	0.16

E. Recommending TV Series Using ANFIS

This system calculates the probable rating of a medium a user might give using Adaptive Neuro Fuzzy Inference System. It will analyze all the TV series ratings and every analysis is done only within the users of any particular cluster. So, calculation overhead is significantly lower compared to whole data set. So, every cluster computed before, contains its own ANFIS structure. This ANFIS has two phases to calculate to train and calculate the estimated rating.

1) Traning Part:

The system builds an ANFIS for each of the previous clusters and each ANFIS trains itself only based on the data within the cluster. In order to train our system, for each rating given by a user it takes 20 column as input data. The first 18 column contains the preferability for each genre of the user. If the rated TV series does not satisfy any particular genre the preferability for that genre is 0. The 19th column contains the overall desirability of that TV series calculated using FIS. The final column contains the original rating given by the user. Because we believe that a user's rating on a TV series is not only based on its desirability but also how much the user prefers the genre of that TV series.

To generate the FIS structure for the ANFIS our system uses SUB Clustering method instead of Grid Pattern method because our input data contains 19 input data and 1 training data. If we wanted to generate only 5 FIS structure for each of the input the total iteration to calculate FIS using Grid Pattern Method will be $5^{19} = 1.9 * 10^{13}$ which is not feasible for our particular type of training data.

Hybrid method was used for training the ANFIS structure. For example, we consider user how rated Game of Thrones, South Park and Planet Earth II. We are only going to consider 5 genres out of 18. TABLE XXI lists genre of three TV series and TABLE XXII shows their desirability. TABLE XXIII shows a user's rating for those TV series. TABLE XXIV shows that user's preferability for 5 genres. The training input of our data contains three sections. The first section contains the genre preferability of the user for that particular TV series. As for the first TV series Game of Thrones only satisfies 2 genre Adventure and Drama so Action and Adventure genre preferability of the user is considered here but Crime preferability is considered as 0. The second section contains the Desirability of the TV series. The third section contains original rating of the TV series given by the user. This section helps to train the ANFIS in training part and checks the validity of the result in testing part. TABLE XXV shows the input format for training data. This table was created from data listed at TABLE XXI, TABLE XXII, TABLE XXIII and TABLE XXIV.

TABLE XXI. TV SERIES GENRE

	Genre 1	Genre 2
Game of Thrones	Adventure	Drama
South Park	Comedy	Animation
Generation Kill	Drama	War

TABLE XXII. DESIRABILITY

	Desirability
Game of Thrones	0.73
South Park	0.61
Generation Kill	0.55

TABLE XXIII. RATING BY USER 1

	Rating
Game of Thrones	10
South Park	7
Generation Kill	8

TABLE XXIV. PREFERABILTY OF USER 1 FOR EACH GENRE

Adventure	Animation	Comedy	Drama	War
0.75	0.56	0.21	0.61	0.45

TABLE XXV. ANFIS TRANING INPUT

Input						Output
Preferability					**Desirability**	**Rating**
Adve nture	**Ani mati on**	**Comedy**	**Drama**	**War**		
0.75	0	0	0.61	0	0.73	10
0	0.56	0.21	0	0	0.61	7
0	0	0	0.61	0.45	0.55	8

2) Testing Part:

The validity of our system was checked using different user of the same cluster who have not participated in our training phase . The input format for testing is same as training but this time the final rating column will check the deviation of our system output from the original rating. Fig. 8 shows our ANFIS model structure.

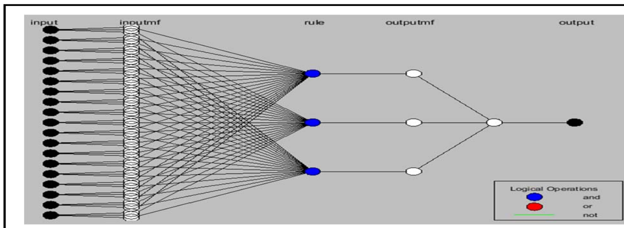

Figure 8. ANFIS Model Structure

IV. RESULT ANALYSIS

For system testing we used 2000 MovieLens [4] users 276027 ratings of 27278 movies to calculate genre preferability per user and then dividing them into 10 clusters. Then we used 579 IMDB [5] users' movie data to assign them to their suitable cluster. Then we used 2470 TV series data and calculated each TV series desirability. Then we used some users tv series information training data and testing data for error calculation.

For example, from cluster 3 we used 1088 training data and 617 testing data. TABLE XXVII shows a comparison between

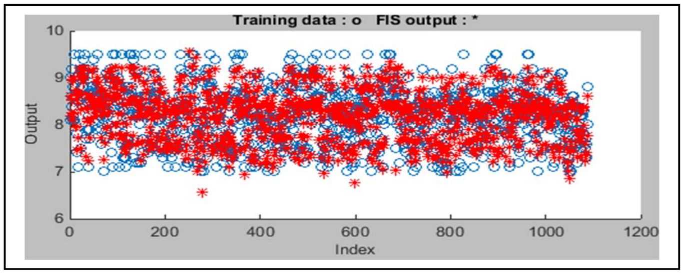

Figure 9. Training Data and FIS Output for Cluster 3

the real rating given by a user and the predicted rating given by the system

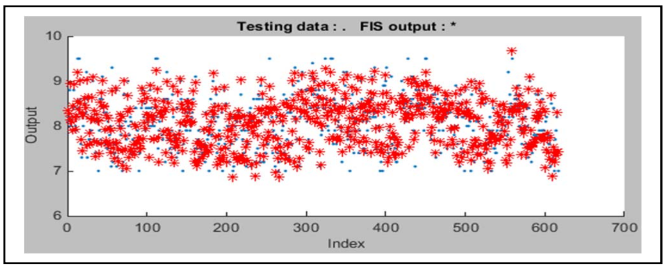

Figure 10. Testing Data and FIS Output for Cluster 3

.

TABLE XXVI. PREDICTATED RATING AND ERROR FOR USER1

	Predicted Rating	Actual Rating	Error
Game of Thrones	8.73	9	0.27
South Park	7.29	7	0.29
Generation Kill	7.44	8	0.66

Average error for training data and testing data for cluster 3 is 0.2517 and 0.27255 respectively.

In another example, from cluster 7 we used 1580 training data and 465 testing data. Average error for training data and testing data for cluster 7 is 0.32214 and 0.32104 respectively.

This shows that our system can predict users rating with a very low amount of error for most of the time.

V. CONCLUSION

The system developed in this paper is the first TV series recommendation system that consides the number of episodes of TV series as an input. Recommending TV series should depend on the number of episodes of that TV series because the length of a TV series has significantly higher impact on a new consumer while looking for a new TV series. Also, a user does not watch many TV series compared to movies and since movies and TV series are similar products, we have decided to use movie data as the initial data of our system. If the database changes it is highly unlikely to have any effect on our result since the membership functions are defined by keeping in mind the real-world conditions that a user considers while choosing a TV series. The result is promising as the average deviation from the actual rating is significantly lower, but more research can improve the result even further.

REFERENCES

[1] "Netflix United Kingdom - Watch TV Programmes Online, Watch Films Online". *Netflix.com*. N.p., 2017. Web. 3 Jan. 2017.

[2] "Watch TV And Movies On Xbox, PS3, Apple TV, And More | Hulu". *Hulu*. N.p., 2017. Web. 3 Jan. 2017.

[3] "Welcome To Prime Video". *Primevideo.com*. N.p., 2017. Web. 3 Jan. 2017.

[4] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=<http://dx.doi.org/10.1145/2827872>

[5] "Imdb - Movies, TV And Celebrities". *IMDb*. N.p., 2017. Web. 2 Jan. 2017.

[6] Ghosh, S., & Dubey, S. K. (2013). Comparative Analysis of K-Means and Fuzzy CMeans Algorithms. *International Journal of Advanced Computer Science and Applications*, 4(4), 35-39

[7] Tsoukalas, L. H., & Uhrig, R. E. (1997). *Fuzzy and neural approaches in engineering*. New York: Wiley.

[8] Hsu, S. H., Wen, M., Lin, H., Lee, C., & Lee, C. (n.d.). AIMED- A Personalized TV Recommendation System. *Interactive TV: a Shared Experience Lecture Notes in Computer Science*, 166-174. doi:10.1007/978-3-540-72559-6_18

[9] Antoine Pigeau, Guillaume Raschia, Marc Gelgon, Noureddine Mouaddib, R´egis Saint-Paul. A Fuzzy Linguistic Summarization Technique for TV Recommender Systems. *IEEE International Conference of Fuzzy Systems (FUZZ-IEEE'2003)*, May 2003, United States. pp.743-748, 2003.

[10] Eyjolfsdottir, Eyrun A., Gaurangi Tilak, and Nan Li. MovieGEN: A Movie Recommendation System (2008): n. pag. Print.

[11] Siddiquee, M. R., Haider, N., & Rahman, R. M. (2015). Movie Recommendation System Based on Fuzzy Inference System and Adaptive Neuro Fuzzy Inference System. *International Journal of Fuzzy System Applications, 4*(4), 31-69. doi:10.4018/ijfsa.2015100103

[12] Cao, Y., & Li, Y. (2007). An intelligent fuzzy-based recommendation system for consumer electronic products. Expert Systems with Applications,33(1), 230-240

[13] Bhatnagar, S., Gopal, L., Velusamy, S., & Varadarajan, S. (2008). An efficient ad recommendation system for TV programs. Multimedia Syst., 14, 73-87.

[14] Janela, M. (2013, September 04). Breaking Bad cooks up record-breaking formula for GUINNESS WORLD RECORDS 2014 edition. Retrieved from http://www.guinnessworldrecords.com/news/2013/9/breaking-bad-cooks-up-record-breaking-formula-for-guinness-world-records-2014-edition-51000

[15] Swatman, R. (2016, September 21). Game of Thrones wins three Emmys and breaks two world records. Retrieved from http://www.guinnessworldrecords.com/news/2016/9/game-of-thrones-wins-three-emmys-and-breaks-two-world-records-444586

[16] 101 Best Written TV Series. (n.d.). Retrieved from http://www.wga.org/writers-room/101-best-lists/101-best-written-tv-series/list

[17] List of most watched television broadcasts in the United States. (n.d.). Retrieved from https://en.wikipedia.org/wiki/List_of_most_watched_television_broadcasts_in_the_United_States