**1. Introduction:**

There are two types of clustering hard clustering and fuzzy clustering. In hard clustering data is divided into distinct clusters, where each sample belongs to specific cluster. Kmeans clustering (Singh, & Yadav, 2013) is type of hard clustering algorithms. Samples in these algorithms belong to distinct cluster or class and these algorithms do not tell us about relationship between samples and other clusters. Samples may belong to more than one class sometimes. This can be easily achieved using Fuzzy clustering which is based on Fuzzy logic. Fuzzy logic is an approach to computing based on degree of truth rather than the usual true or false (1 or 0) Boolean logic. We have used Fuzzy C-means Clustering (Gosaina, A., & Dahiyab, 2016; Tayal, Ahuja, & Chhabra, n.d.) assigns samples to more than one cluster based on membership value for each cluster.

Datasets for this experiment have been collected from IMDb website and Kaggle dataset (IMDb 5000+ Movie Dataset). We have created custom web crawler script for extracting data from Imdb website. A Web Crawler also known in other terms like ants, automatic indexers, bots, web spiders, web robots or web scutters is an automated program, or script, that methodically scans or crawls through web pages to create an index of the data it is set to look for. This process is called Web crawling or spidering.

Data is collected in different sources – IMDb and Kaggle movie database. Features extracted are movie name, director name, top 3 actors name, top 7 genres, budget, gross, duration, color, year, Imdb rating, language, country. The Internet Movie Database, abbreviated IMDb, is an online database of information related to films and television programs, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews. As of June 2017, IMDb has approximately 4.4 million titles (including episodes), 8 million personalities in its database, as well as 75 million registered users. So, it very good source of data. Kaggle is online data science competition site which also provides data under “Open Database License”. Python's elegant syntax, dynamic typing, and wide range of libraries is the reason we chose this language.

**2.** Preliminaries**:**

**2.1 Clustering:**

Clustering is a process of making group of anything possible. In data mining, clustering is technique of dividing samples into classes or clusters such that intra-cluster similarity is increased and inter-cluster similarity is reduced. There are many measure of similarity like distance, connectivity, intensity or density, etc.

**2.2 Fuzzy Logic:**

Fuzzy logic is an approach to computing based on degree of truth rather than the usual true or false (1 or 0) Boolean logic. The term fuzzy logic was introduced with the 1965 proposal of fuzzy set theory by Lotfi A. Zadeh (1965). Fuzzy logic has been applied to many fields, from control theory to artificial intelligence.

In fuzzy clustering sample can belong to more than one cluster, depending on the membership value of sample. Membership value represents strength of association between sample and a particular cluster. Therefore, we can define Fuzzy clustering as process of assigning these membership values, and then using them to assign samples to one or more clusters.

**2.3 K-means Clustering:**

K-Means clustering (Singh, & Yadav, 2013) intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly k different clusters of greatest possible distinction. The best number of clusters k leading to the greatest separation (distance) is not known as a priori and must be computed from the data. The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function:

*Equation-1: Kmeans*

**2.4 Fuzzy C-Means Clustering:**

In (Ross, 2004), Fuzzy C-Means (FCM) is described as a method of clustering which allows one piece of data to belong to two or more clusters by minimizing the following objective function:

*Equation-2: Fcm*

* m is any real number greater than 1
* uij is the degree of membership of xi in the cluster j
* xi is the i-th of d-dimensional measured data
* cj is the d-dimension center of the cluster j
* ||\*|| is any norm expressing the similarity between any measured data and the center.

Equation-2 is iteratively optimized alongside updating membership value uij and the cluster centers cj:

*Equation-3:*

*Equation-4:*

Fuzzy C-means Clustering is modified form of k-means clustering where we take in membership value uij of sample for each cluster.

**3. Proposed Approach:**

**3.1 Data Collection and Cleaning:**

This is very crucial step of any data mining analysis. Some important features that extracted from each movie are as follows - *movie name, director name, top 3 actors name, top 7 genres, budget, gross, duration, color, year, Imdb rating, language, country*.

Data was cleaned before using because noisy/missing data can impose threat on final result.

* Missing data of ‘duration’ filled with mean of all durations of movies.
* All leading and trailing whitespaces were removed from every value.
* Genres had lot of missing data.

Top two genres were used because other genres had more than 35% of missing data as shown in figure-1.

**3.2 Preparing Data:**

We have to convert every non-numeric data to numeric data. For each feature all, unique values were collected and they were numbered from zero to number of unique values minus one. These numerical alternatives were used for clustering.

We used sci-kit learn scale function to standardize the dataset. It centers to the mean and component wise scale to unit variance. A feature has a variance that is orders of magnitude larger than others, it might dominate the objective function.

**3.3 Features used for clustering:**

|  |  |
| --- | --- |
| S. No. | Feature |
| 1 | Director Name |
| 2 | Actor 1 Name |
| 3 | Actor 2 Name |
| 4 | Actor 3 Name |
| 5 | Duration |
| 6 | Year |
| 7 | Imdb |
| 8 | Country |
| 9 | Genre 1 |
| 10 | Genre 2 |

Table 1: Selected Features

Features given in Table-1 are selected as features because these are the most favorable factors on which people decide to watch movies. They select movies according to their favorite director or actors. Other selects movies by their preference of particular time period or genres or length of movie and many times IMDb rating.

**3.4 Flow Chart of process:**

Training Data with numeric and text values

Training data with numeric values only

Training Data with 10 movie features

Cluster Centers of Fuzzy C-Means Clustering

Cluster Centers of K-Means Clustering

Scatter Plot of Clusters

3-Dimensional Data

Outlier removed dataset

Figure 1: Flow Chart of process

**3.5 Clustering:**

Data extracted have both text data (e.g. director’s name, actors’ names, genres, etc.) and numeric data (e.g. Imdb rating, duration, etc.). So, we need to handle non-numeric data otherwise we will not get desired result. We will assign values from 0 to N in each column according to set of values in that column, where N is number of unique values in a column.

|  |  |  |
| --- | --- | --- |
| No. of Features | No. of samples |  |
| 10 | 4317 |  |

Table 2: Shape of data used for both clustering algorithm.

Table 4: Parameters for K-Means

Now this 10-dimensional data cannot be used for visualization, it is important to reduce its dimensions. We can visualize this data in 3-dimentions using matplotlib library easily. So, we apply Principal Component Analysis (Shlens, 2005) to reduce it into three dimensions.

Now we can apply FCM and K-means.

**3.5.1 Parameters for FCM and KMeans:**

Table-3 and Table-4 shows parameters used for Fuzzy C-Means and K-means clustering respectively. Initial fuzzy c-partitioned matrix is the initial value of uij which is randomly taken. Similarly, initial centroids for K-Means clustering is taken random initially. ‘Maximum number iteration’ is the maximum number of times both algorithm should iterate.

|  |  |  |
| --- | --- | --- |
| Parameters |  | value |
| Clusters |  | 10 |
| Number of different centroids |  | 10 |
| Maximum Number of iterations |  | 1000 |
| Error Rate |  | 0.0001 |
| Initial centroids |  | random |

Table 3: Parameters for FCM

|  |  |
| --- | --- |
| Parameters | value |
| Clusters | 10 |
| m (exponential applied to membership function) | 2 |
| Maximum Number of iterations | 1000 |
| Error Rate | 0.0001 |
| Initial fuzzy c-partitioned matrix | random |

**3.6 Silhouette Index and Davies–Bouldin Score:**

Silhouette Score (Liu, Li, Xiong, Gao, & Wu, 2010) and Davies–Bouldin (Liu, Li, Xiong, Gao, & Wu, 2010) it measures the quality of clusters produced by clustering algorithms. The silhouette width is calculated by averaging all the silhouette values for each pathway, where the silhouette value is calculated using the following function:

*Equation-5:*Silhouette Index

*a*(*i*) is the distance of *Xi* to its own cluster, which is defined as the average distance of *Xi* to all the other samples in its own cluster. *b*(*i*) is the distance of *Xi* to its closest neighbouring cluster, which is defined as the average distance of *Xi* to all the samples in its closest neighbouring cluster

Davies–Bouldin is the ratio of intra-cluster distance to inter-cluster distance. This means we need to minimize the objective function so that intra-cluster distance is minimum and inter-cluster distance is maximum.

*Equation-6:*Davies–Bouldin Score

*Equation-7: Diameter*

*Equation-8: Euclidean Distance*

with ni the number of points and zi the centroid of cluster ci.

**3.7 Outlier Detection and Removal:**

Outlier removal is necessary because outlier can degrade our result. We have used Local Outlier Probabilities (Kriegel, Kröger, Schubert, & Zimek, n.d.).

PLOF is Probabilistic Local Outlier Factor of an object o D w.r.t. a significance , context set S(o) D, where erf is Gaussian error function and D is set of *n* samples.

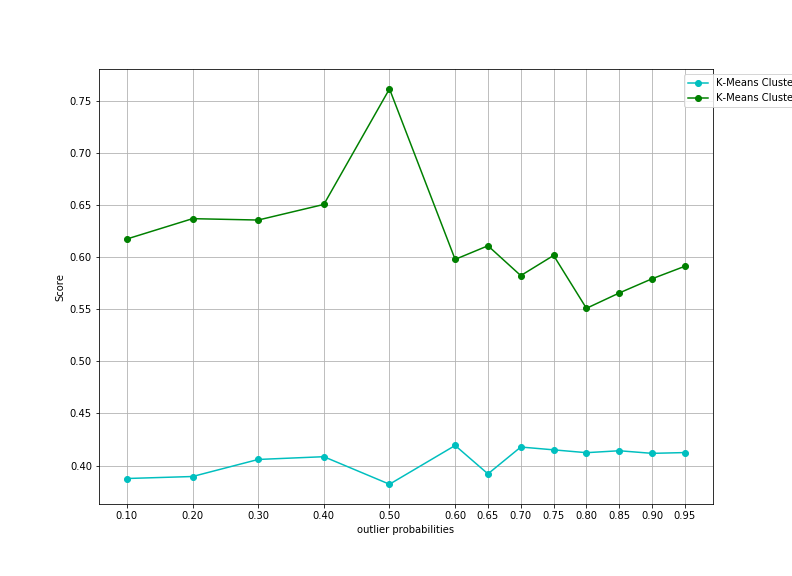
****

Figure 2: Silhouette(green) and Davies-Bouldin(blue) for probabilities 0.1 to 0.95 for Kmeans

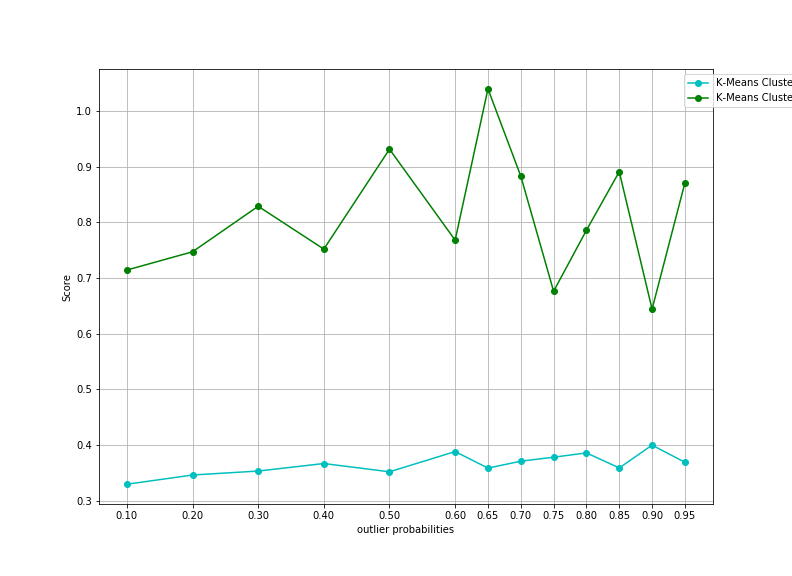
****

Figure 3: Silhouette(green) and Davies-Bouldin(blue) for probabilities 0.1 to 0.95 for FCM

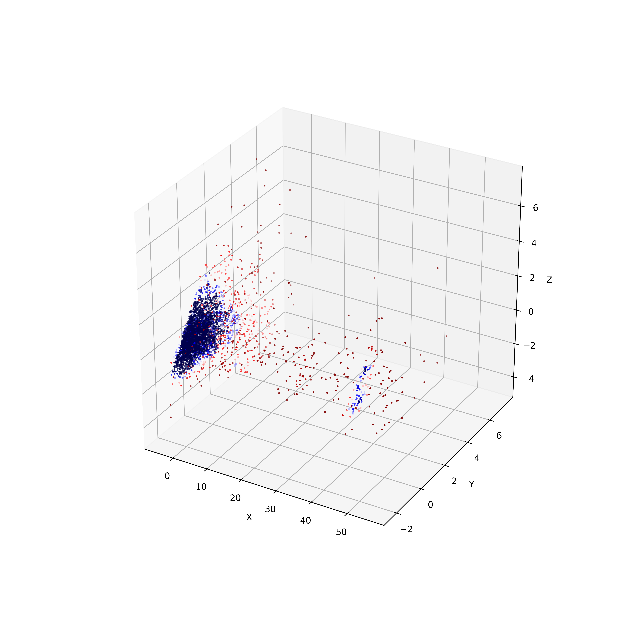
**4. Visualization:**

Figure 4: Samples with outlier probabilities

Figure-4 shows distribution of samples with outlier probabilities from 0(blue) to 1(red). Figure-4 shows distribution samples with Kmeans clusters and Figure-5 shows distribution samples with FCM clusters with their respective centers, with outliers removed using Local Outlier Probabilities.

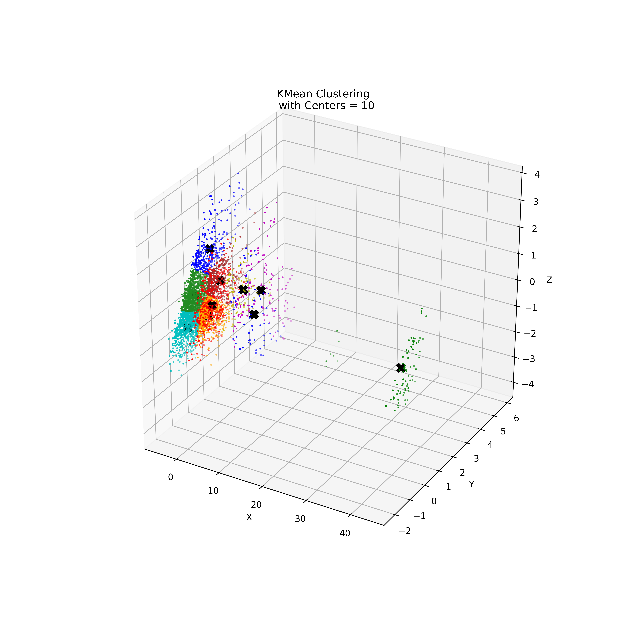
****

Figure 5: FCM clusters

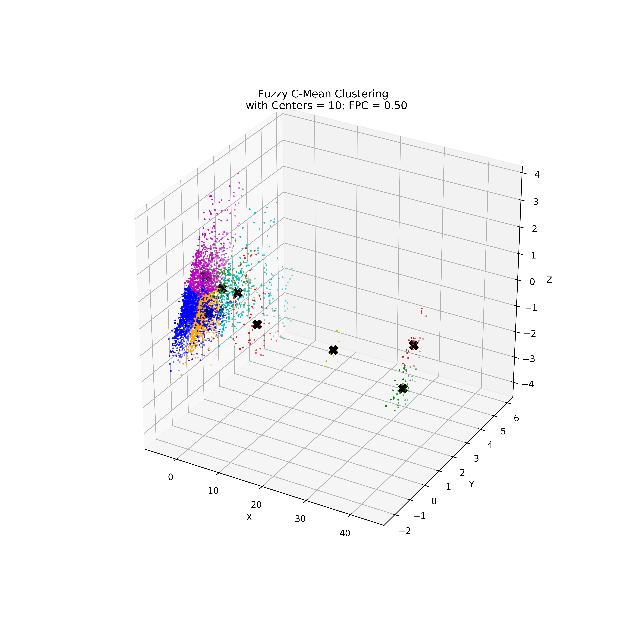


Figure 6: Kmeans clusters

**5. Result:**

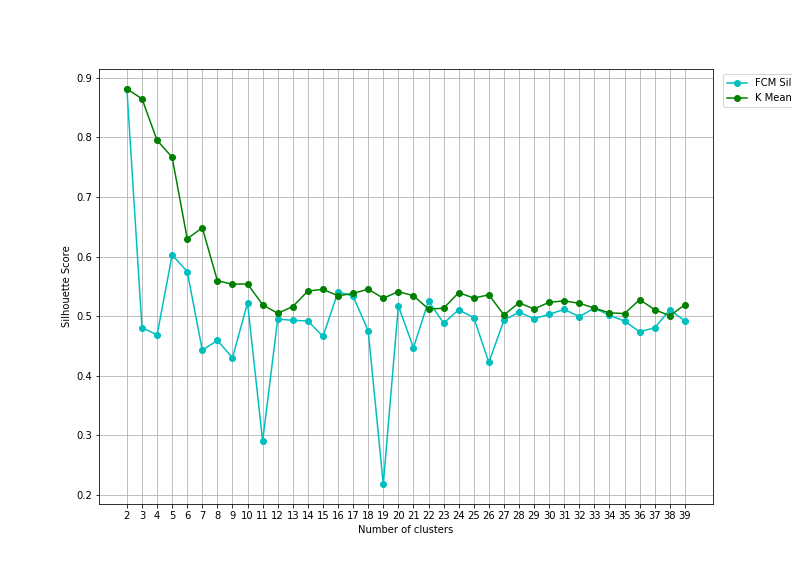
****

Figure 7: Silhouette Score Kmeans and FCM

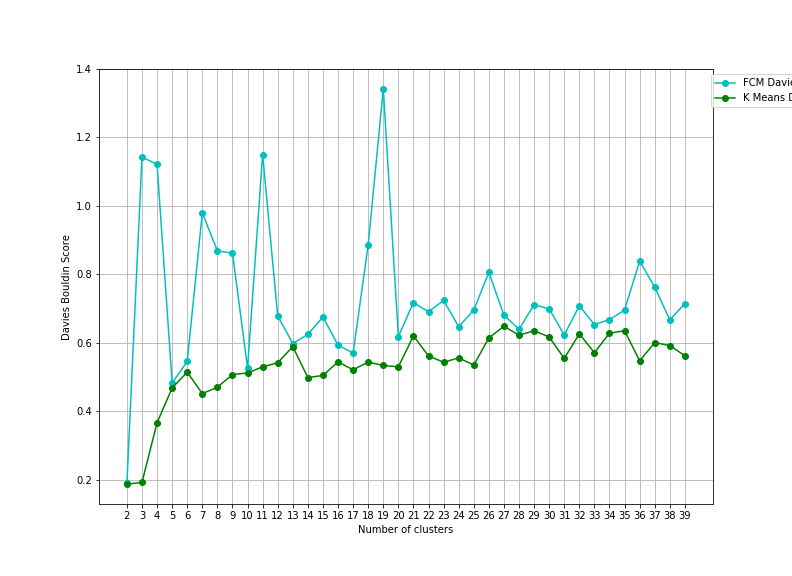
****

Figure 8: Davies-Bouldin Score Kmeans and FCM

As we can see Silhouette Score is mostly higher for different number of clusters in Kmeans than FCM, this suggest Kmeans produce better clusters. Also, Davies–Bouldin Score is lower for Kmeans which implies same result. Time complexity of FCM is O(ndc2i) and of K-means is O(ndci) where n = number of samples, d = number of dimensions, c = number of clusters and i = number of iterations.

|  |  |  |
| --- | --- | --- |
| Algorithm | No of executions | Time(seconds) |
| FCM | 1 | 0.77 |
|  | 2 | 0.79 |
|  | 3 | 0.70 |
| K-Means | 1 | 0.16 |
|  | 2 | 0.17 |
|  | 3 | 0.16 |

Table 5: Time elapsed

So, we can see K-means had advantage over Fuzzy C-Mean. We have achieved time elapsed for Kmeans faster to that of FCM and better external validation scores for Kmeans. Therefore, we can conclude Kmeans produce better clusters than Kmeans for movie database recommendation system.

**6. Reference:**

**1.** Tayal, D. K., Ahuja, L., & Chhabra, S. (n.d.). Word Sense Disambiguation in Hindi Language Using Hyperspace Analogue to Language and Fuzzy C-Means Clustering.

**2.** Singh, S. P., & Yadav, A. (2013). Study of K-Means and Enhanced K-Means Clustering Algorithm. International Journal of Advanced Research in Computer Science Volume 4, No. 10.

**3.** Pranav, A., & Chauhan, S. (2015). Efficient Focused Web Crawling  
Approach for Search Engine. IJCSMC, Vol. 4, Issue. 5.

**4.** Kriegel, H.P., Kröger, P., Schubert, E., & Zimek, A. (n.d.). LoOP: Local Outlier Probabilities. Institut für Informatik, Ludwig-Maximilians Universität München.

**5.** Gosaina, A., & Dahiyab, S. (2016). Performance Analysis of Various Fuzzy Clustering Algorithms: A  
Review. 7th International Conference on Communication, Computing and Virtualization.

**6.** IMDb 5000+ Movie Dataset. Retrieved from <https://www.kaggle.com/nazimamzz/imdb-dataset-of-5000-movie-posters/data/>

**7.** Shlens, J. (2005). A Tutorial on Principal Component Analysis.

**8.** Liu, Y., Li, Z., Xiong, H., Gao, X., & Wu, J. (2010). Understanding of Internal Clustering Validation Measures. IEEE International Conference on Data Mining.