Movie Recommendation System

Angel Shrestha, Erosha Pande, Shaline Kocherla, Tarun Kumar Devan

CSCE 5380 Final Project Report

Department of Computer Science and Engineering

University of North Texas

**Abstract**— Movies are a popular way to earn money these ways. And among the increasing popularity of the movies, online movies are ever-growing. And with the increasing number of movies, choosing between them is hard for a user while interacting with the internet hub. So, we propose a movie recommendation system based on users rating distinguished by genres. We implement our own algorithm for predicting the topmost genre preferred by a particular user. By combining it with clustering techniques and data mining methods, we could predict the system’s accuracy as well. We were able to compare the techniques using our own algorithm but this prediction could be further processed for its validity.

Keywords— Movie recommendation system, K-Means Clustering, Spectral Clustering, Mean-Shift clustering, Comparison analysis

1. Introduction

Entertainment Industry, specifically movie business is one of the most money-making industry in the word. We are considering this movie business as our application domain. We are specifically targeting movie theater websites. Every year thousands of movies are produced and released so audiences have tons of options to choose from. There are tons of movie rating websites however most of them do not provide recommendations to the users based on their preferences. The choice of movies is strictly subjective and varies from person to person. To solve the problem of finding suitable recommendation based on individual preferences, we have based our project which provides suitable options for movie buffs to help them select what to watch next based on their favorite movie genre.

To solve the problem of finding suitable recommendation for a particular user we have tried to implement a movie recommendation system to find the best options for them based on the movie genres they previously watched and had given ratings for those movies.

1. Formal Problem Definition

Our project strives to get the five genres the customers are most interested to watch next based on the movies they have rated.

1. Methodology

Our data mining plan involved collection of the movie rating dataset. After the collection of the required dataset we cleaned the data set. We used the users to form clusters based on common interest in movie genres. Based on those clusters we found the five most popular genres of movies for that particular cluster. Then predicted the five genres that the users would be interested in watching. The methodology is elaborated in the methodology section.

1. Dataset

For our analysis we used datasets available from MovieLens which is a development dataset. It has data for 100004 ratings across 9125 movies and for 671 users. For our project we utilized two datasets – Movie Dataset and Rating Dataset. The various attribute of these datasets is represented in table 1. Since the attributes we wanted to use for our project were UserID and genres, we merged the two datasets by using MovieID as the common attribute between them.

By using these two attributes we were able to get clusters based on users who preferred similar genre of movies. We didn’t consider the attributes such as timestamp and ratings because we were solely focused on clustering the users based on their preference of movie genres. Hence, in the merged dataset which is our final dataset, UserID, MovieID, genre are the useful attributes.

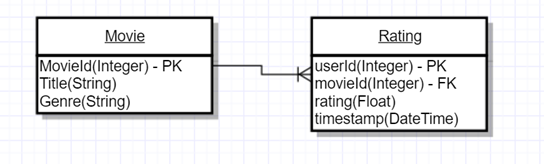
We also created a table to count the responses for each movie IDs. In this table the MovieIDs are used as our table Header/attribute. Each row in this table are the user ID. If the users represented by UserID has given response for a particular MovieID the count will be incremented for that MovieID say 1. The dataset contained a lot of movies which had no genres which were not helpful for our analysis. Also, there were a lot of “NaN” values which represented cases when the users didn’t rate that particular movie. So, for this, we have performed data cleaning in the dataset. We replaced all the “Nan” values with 0.0 values and removed useless data belonging to “no genre” category.

|  |  |
| --- | --- |
| Movie Dataset | Rating Dataset |
| Attributes:  1. movieId: Unique Id for each movie  2.Title: Name of the movie  3.Genres: Each movie can have more than one genre | Attributes:  1. userId: Unique Id for users  2. movieId: Foreign key for the above dataset  3. Ratings: 0.5-5.0 with .5 increment basis  4. Timestamps: in seconds since midnight |

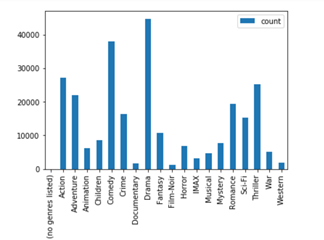
Table 1 Datasets

1. Exploratory Analysis

To give good idea of dataset used, we have given an UML diagram to visually represent how we used MovieID (Foreign Key) to merge the two datasets.

Fig. 1 UML diagram

We have used a bar plot to get the idea of which should the most popular genres in most of the clusters. Here, we can clearly see that drama is the most popular genre in our dataset. So, drama can be assumed to be preferred by most of the users.



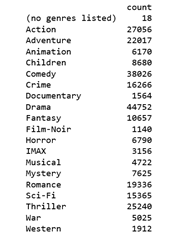


Fig. 2 Bar graph representing the distribution of ratings across the genres

We have used scatter plot to represent our dataset. Since there were tons of attribute we used Principal Component Analysis(PCA) for dimensionality reduction. Principal Component Analysis (PCA) is a dimension-reduction tool that can be used to reduce a large set of variables to a small set that still contains most of the information in the large set. The x-axis and y-axis represent PC1 and PC2 respectively.

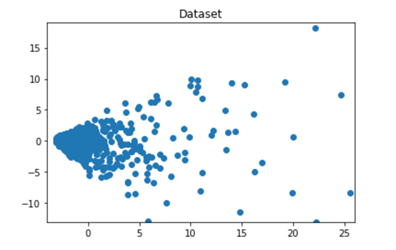


Fig. 3 Scatter plot of our dataset

The scatter plot above clearly shows how compact the data points in our dataset are. This helped us a lot to determine which clustering algorithm can work well with our dataset. We chose three clustering algorithms - K-Means, Spectral Clustering and Mean Shift which have been elaborated in the Methodology section of this report. We looked around different related research works and paper to find which clustering method would be the best one to choose from. From the research we conducted, we found that K-Means was the most popular one that could help in recommendation systems as the points plotted would be compact in most cases.

And we chose Spectral Clustering in the starting because it also takes number of cluster as an input. But as we went on researching more and more about this type of clustering, an interesting fact came up in front of us. This type of clustering doesn’t quite work well with the compact dataset. So, we wanted to try it out and learn something out of it. If we are learning what we should do, we should also learn what not to do. Only then our knowledge would be complete on that subject. While finding out about Spectral Clustering, we also got to know that Mean-Shift clustering also doesn’t work well with the compactness of the dataset but with a totally different reason which we would find out as we go on with this project.

1. Related Work
2. Movie Recommendation using Unrated Data

This paper is using Movie dataset on which we are working. In this IEEE paper they are taking the unrated data and then giving predictions by applying their own method and later Pearson Correlation Coefficient (PCC) method is done so that the similarity between users is known and then matrix factorization is developed to get genre diversity and obtain the recommendations. Using Movie Frequency-Inverted Genre Frequency (MF-IGF) is the formula used in this paper here, the genre selected is directly proportional to list of movies of a particular genre in users watched movies. Also it is inversely proportional to the number of times the particular genre is repeated in the dataset. So if the users mostly views movie is sparse then by applying the above formula we can get the users actual preferences

Next is using PCC to find similarity, here PCC is used as it considers different users that have different genre liking. The MF-IGF gives the users average genre preference. Normally the low-ranking matrix factorization approach can approximate rating matrix. Using the singular value Decomposition and matrix factorization methods the genre preferences is found which helped in the movie recommendation. Here we are making sure that the specific user interest is close to the average taste of k-nearest users.

Now the movielens dataset is considered and it is divided into training and testing sets. And the training set is used in the algorithms and testing dataset is used to measure the performance of the training set here, we are using RMSE.

The above is implemented in python and the pre-processing of the data is done in MATLAB the algorithms implemented are AVGB, SVD, CSVD. The below figure shows the results of alpha which directly controls the degree of incorporation against RMSE.

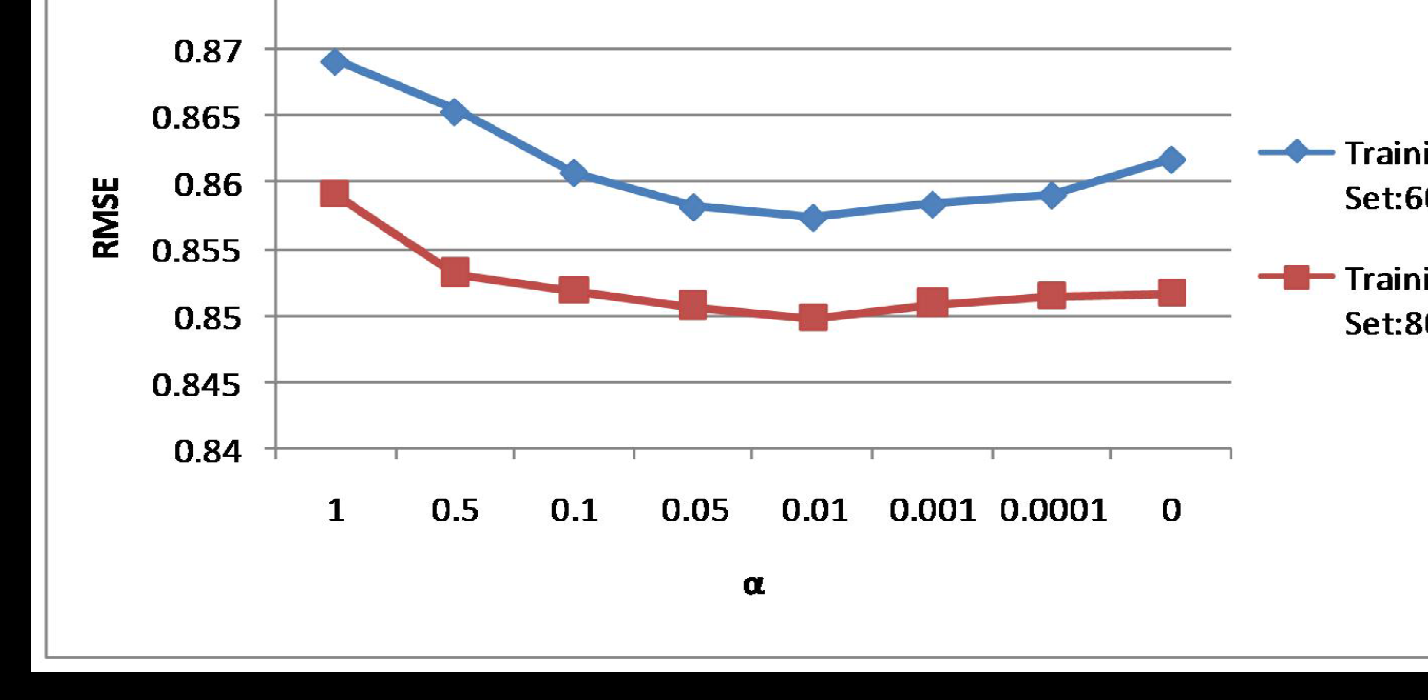


Fig. 4 RSME vs. alpha

From the figure we can conclude that alpha effects the results so, it can be used to improve recommendation system proposed using genres. k is a parameter which tells how many neighbors are present, tis parameter also effects the performance of RMSE. Below is the graph of k versus RMSE

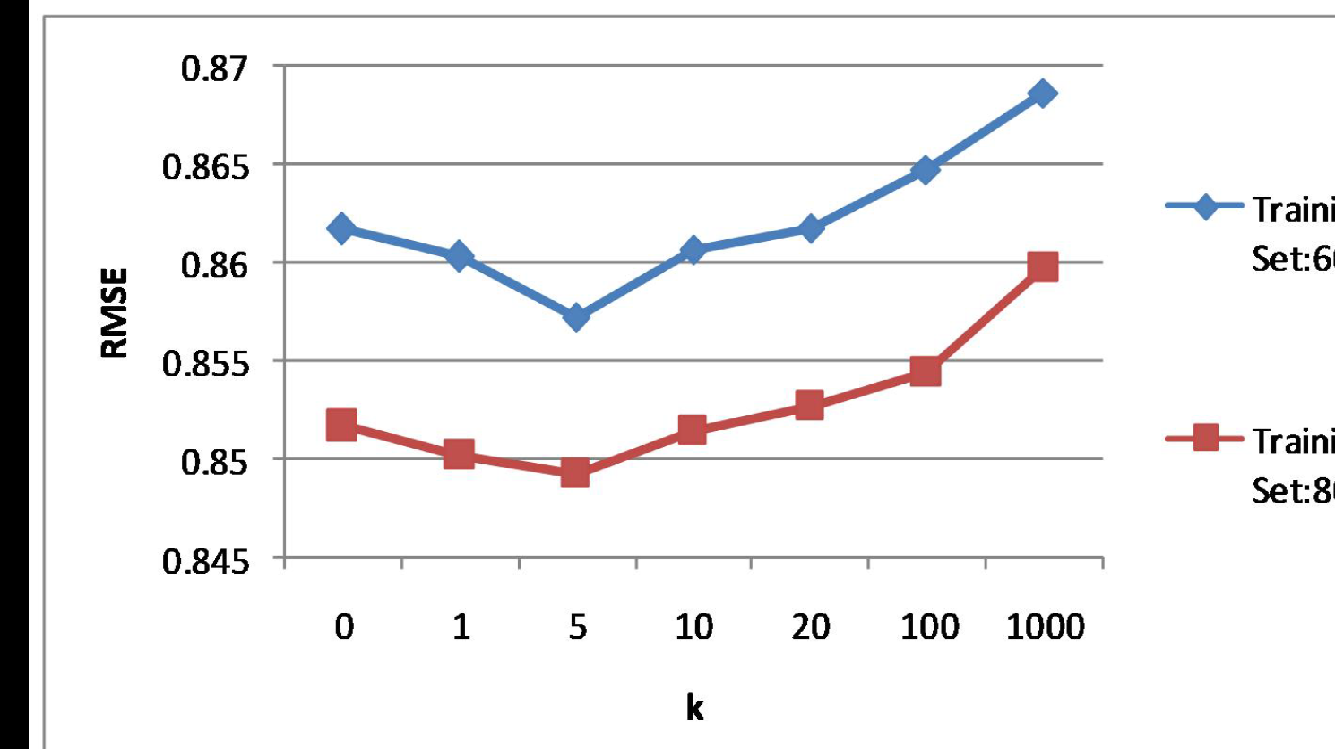


Fig. 5 RSME vs. k

The genre is the main attribute used to run this model and recommend movies. In this the rating between users and movies, predict the rating for each user and sort them. Then we choose the top N movies having high prediction rating these are recommended to the user. Analyzing the results we can tell that this model is better than all the baseline algorithms and the genre diversity is improved. The above model can also be used to recommend books music etc.

1. Book Recommendation System Based on Combine Features of Content Based Filtering, Collaborative Filtering and Association Rule Mining

In this paper a recommendation system on books is proposed based on the users’ interest. In this content filtration collaborative filtration, association mining is used to obtain best recommendations based on the buyer’s choice

In the content recommendation system web usage mining is considered which stores the user behavior, it also generated information which is more accurate and close to the users.

The content recommendation system uses the content of the users previous buying records it uses content based filtration to separate users. But there are some limitations in the content based filtration like the quality of the content In order to overcome this collaborative filtration system is used this depends on users views. This is generally used in the recommendation system on the web this is also called as social filtering or recommendation system Here, one item is targeted and then similar items are found based on the users rates later the recommendation system is computed. To find similarity in two items each one is considered to be a vector in user space and cosine function is applied between two vectors. In the Association Rule Mining the relation between large datasets in found. Market basket is one of the example here patters are found based on support and confidence. In this paper we will consider customer information table, book information table, category table, order information table and order detail table. The specialty of this paper is the recommendation system proposed works offline and stores all the recommendation in the web profile of the user. This has seven major steps.

Firstly, buyers profile record is considered to find the category of books and the sub category these are used for filtering the transactions as well as content based filtration from here collaborative filtration is performed the output of this is given as input to the intersection box. The output of transaction filtration is association rule mining later considering all inputs the final recommendation is formed.

1. Methodology

Our project’s main goal is to predict the topmost genres that a user would prefer based on the data that we have now. So, we are getting a descriptive pattern in our dataset. For this, we have followed the basic steps of data mining which is shown in the following flowchart.



Fig. 6 Project Flow

1. Data Collection

This proved to be the most important step in our project. Finding the data that could give us better clusters was a challenge as there is very less options to how to get the validity of the clusters. We checked multiple repositories for datasets like KDnuggets, UCI, Kaggle, etc. before finding the one we chose and each one had something or the other as drawback. Like some of them had really less number of instances and couldn’t give us the proper result that we needed. And some of them didn’t have the attributes that we needed. For example, we need ratings that could be related with both the movies and genre but most of them had only either movie and user or movie and rating.

When we came across GroupLens, it proved to be the best one to get the required dataset for our project and the dataset matched what we had in mind. The dataset that we collected is from the website of GroupLens namely <https://grouplens.org/datasets/movielens/>. This dataset describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. This dataset is a development dataset and it might change over time as well. There are two different dataset that could be related to each other to get the movie recommendation. They were ratings.csv and movie.csv.

We need both datasets to get the final output of our project as one of them contains movie and their genre and another one of them has information about users and the movie they have rated and our final goal is to interrelate users with genres.

Hence, from here, we start the programming part of our project.

1. Data Cleaning

We read both the datasets into pandas dataframe so that they would be accessible to us in an easy way. As we got from our exploratory analysis, there are missing values in the form of movies that have no genres listed. And as only a few users have rated certain movies, the correlation table from merged dataset between user and movies in the ratings dataset that we create in the data preprocessing part would have a lot of NaN values.

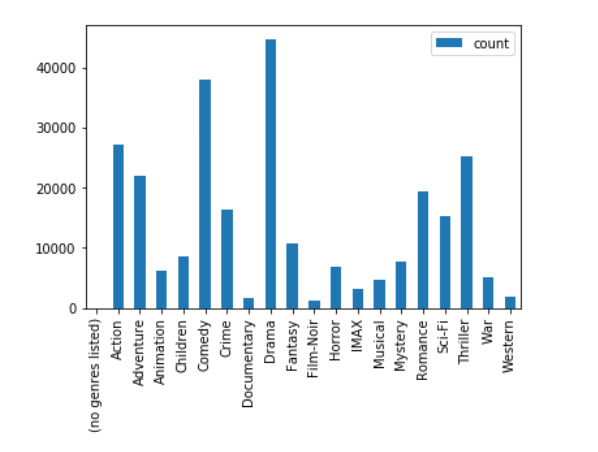


Fig. 7 Bar graph representing the distribution of ratings across the genres

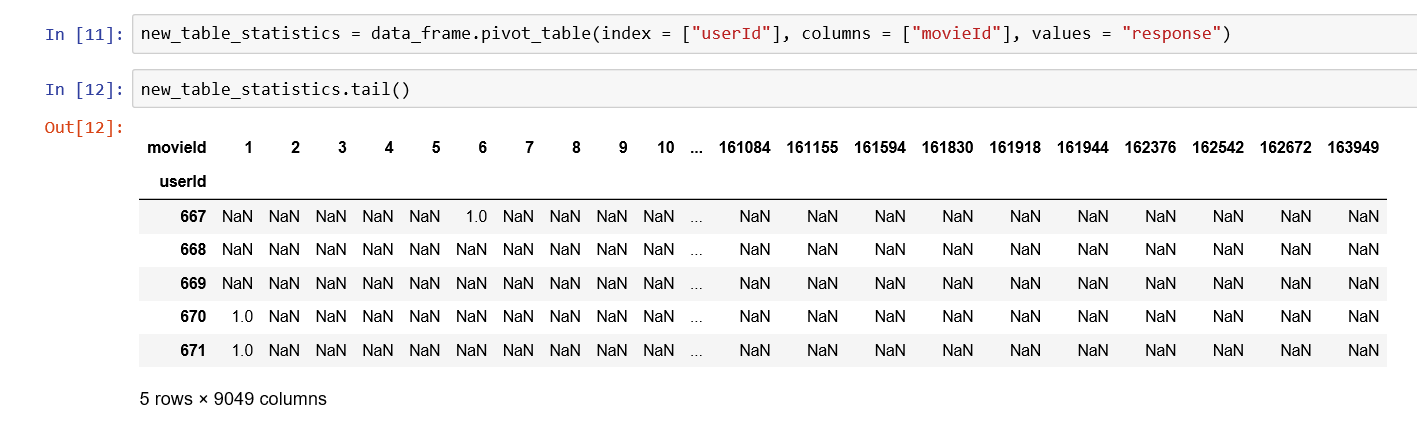


Fig. 8 DataFrame having NaN values

So, we had to perform data cleaning step before continuing with our flow. For this, as the movies that had no genres listed had no contribution in our prediction, we cleaned that part out and removed in from the dataset. But for the ones with NaN values, we replaced it with 0 which gives the same meaning as the ratings are not present. So, this would be our data cleaning part of the project.

1. Data Preprocessing

Now, in this part, we are merging the two datasets movies and rating into one so that we can get the relation between user and movie Id as well as user and genre. And a response column is added so we could plot the correlation between the user and the movies that they have rated.

As we already discussed about cleaning this correlation table, we would get a ready dataset for our further preprocessing. Now, at this part, we use Principle Component Analysis to plot this dataset as a scatter plot which we have used for the exploratory analysis part.

1. *Principal Component Analysis:*

It is a method of representation for compressing a lot of data into something that captures the essence of the original data. It takes a dataset with a lot of dimensions and flattens it to 2 dimensions so we can visualize it. The x-axis which is also known as the PC1 would span the direction of the most variation and y-axis, which is also known as the PC2 would span the direction of the 2nd most variation.

So, it plots the x and y-axis for the correlation table that has a lot of dimension and shows how they show up in the graph and which ones have similar characteristics. So, x-axis shows the movie and user relation that has the most variation and y-axis shows the one which has the 2nd most variation.

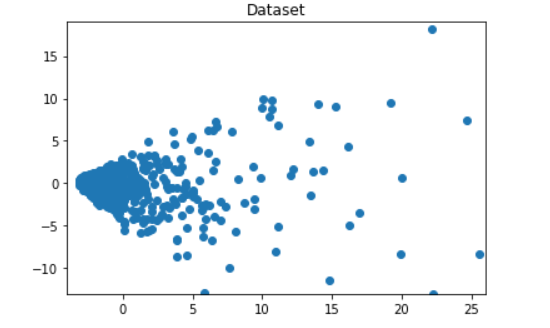


Fig. 9 Scatter plot representing the dataset using PCA

Now, the x and y column that we get from the Principle Component Analysis for the gotten dataset is used in the rest of our project for clustering. The idea is that each dot in the scatter plot represents the user rating for a certain movie and the ones with similar characteristics should cluster. The rest of the project part is discussed in following Data Mining step.

1. Data Mining

As from the exploratory analysis we got, the major method that we used for our project is K-Means clustering technique and for comparison, we have used Spectral clustering technique and Mean-Shift clustering technique.

The outline of this heading would be introduction to K-Means, the project flow and finally the experimental result and comparison with the other methods that we have used.

1. *K-Means Clustering Description*

K-Means clustering is a type of unsupervised learning, which is used when you have unlabeled data. It takes in input number of clusters and divides the dataset into that many number of clusters. It uses centroids and clusters the points comparing the distance between the points and the centroid.

1. *Project Flow*

So, we have used sklearn library for K-means clustering.



Fig. 10 K-Means implementation using sklearn library

But the challenge that we faced before we could start the clustering is to find the optimum number of clusters. We cannot put it randomly as the dataset ranged up to 100004 range. So, we researched on it. The one method that proved to be the most effective is Elbow Method.

**Step 1: Elbow Method**

Elbow Method is a method of interpretation and validation of consistency within cluster analysis designed to help finding the appropriate number of clusters in a dataset. This method looks at the percentage of variance within the cluster to find the consistency in the distribution of data in the cluster. Using this method, we can plot a number of clusters vs. distortion (percentage of variation) graph where there will be a slant slope in the starting and slowly, the slope becomes consistent. The bend where this happens is the optimum number of cluster that we would have to use.

The input for this method would be the range of number of clusters that is used to determine the bend for the given dataset. We used three different range to find the optimum number of clusters.

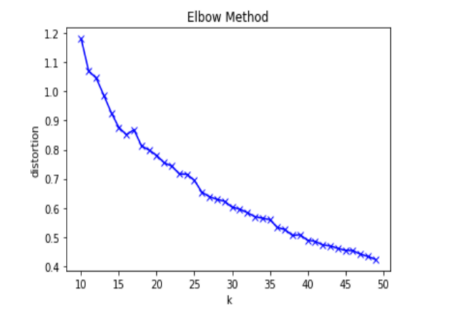
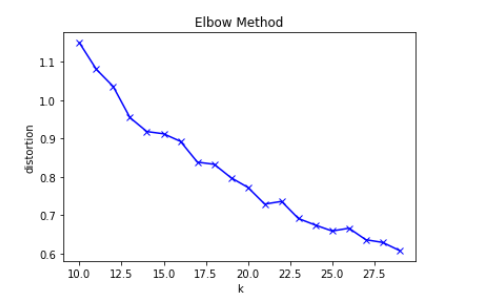
1. 10 to 30

When we used 10 to 30 as a range input for Elbow Method, it couldn’t give us a proper bend as you can see in the figure 11. So, we couldn’t use any value between these range.

1. 10 to 50

When we used 10 to 50 as a range input for Elbow Method, it actually gave us a pretty good result. We got a bend around range of 30 to 35 so, we chose our optimum number of clusters as 35 for the rest of our project. Figure 11 is the graph that we got for the Elbow Method.

1. 10 to 100

When we used 10 to 100 as a range input for Elbow Method, it gave us good result as well just like the one that 10 to 50 range gave. So, for this, we got a bend somewhere in the middle of 40 to 60 as you can see in the figure below. So, we chose the optimum number of clusters got from this range as 50. Below is the graph that we obtained by using 10 to 100 range of number of clusters.

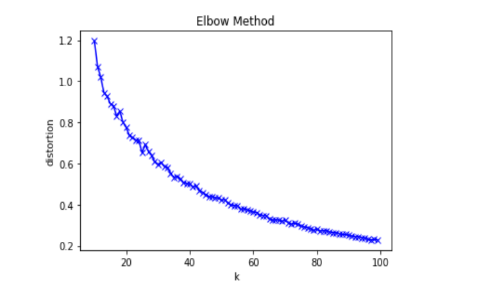


Fig. 11 Elbow method using 10-30, 10-50 and 10-100 range.

According to above result, we used 35 clusters and 50 clusters to carry on with our rest of the project. Now, after finding the optimum number of clusters, we move on to a major part in our project clustering.

**Step 2: Clustering**

Now that we have the optimum number of clusters that we could try our clustering with, we could move on to the clustering part of our project. Let’s recall what we have got till here. We got the PC1 and PC2 which are our two-different principal component axis that helped us in dimensionality reduction as we had loads of attributes. Then, we found out the optimum number of clusters.

After this step, we applied the K-Means Clustering on the correlation table that we got, it had userId and movie Id which are correlated to each other based on which user had rated which movie. And we had added the PC1 and PC2 component that we calculated using Principal Component Analysis. This dataset was fed to the built-in K-Means function provided by sklearn. We have already discussed how the K-Means Clustering works.

So, the output that we get from this function is the prediction of which user will be allocated to which cluster. As the output is a correlation dataset, to perform our final prediction, we had to extract the required dimensions from this table and merge in into the original dataset. After this step, we get the following table.

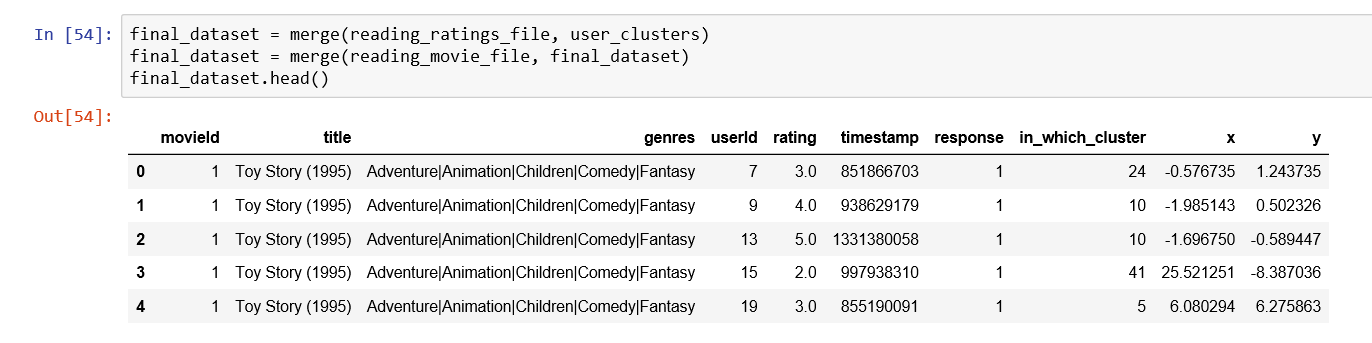
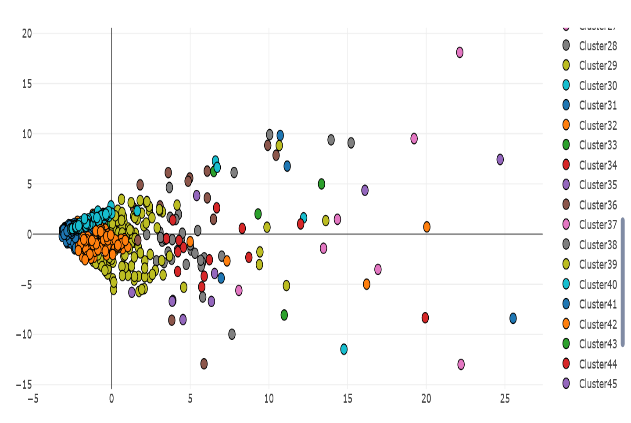


Fig. 12 Final DataFrame that we got as a result for prediction

After this, to visualize and analyze the cluster even more clearly, we plotted the scatter plot assigning different colors to the different colors using go.Scatter. As we used 35 and 50 as number of clusters, the results were as follows:



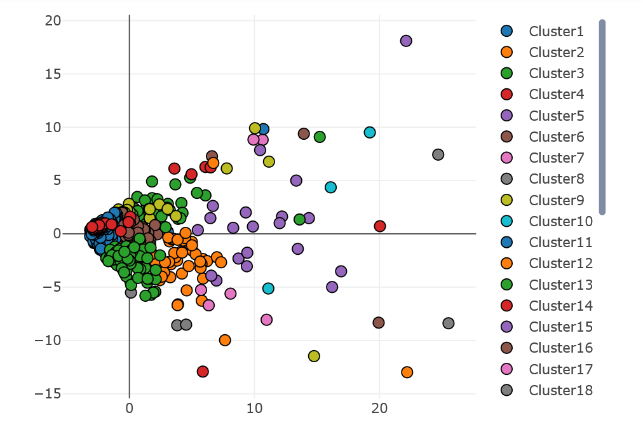


Fig. 13 K-Means Clustering using 50 clusters and 35 clusters.

**Step 3: Getting the generalized most popular genres in the cluster**

This is the major step of our project as this is the part where our goal is attained. This is the part where we find the top 5 most popular genres in a cluster.

Our thought in this part is that the users have been clustered according to the similarity of their preferences based on their ratings. So, if we could find a generalized topmost preference that can be used for all the users that it encircles. And we wouldn’t have to find for the singular user. Hence, we applied an algorithm to get the generalized topmost preference. Now, it could have been done easily with a count of each unique genre but another major challenge that we faced in this project resides here. It was because a single movie could have multiple user. So, the movie that the user has rated wouldn’t just be of one genre but of multiple genre. We have also considered this fact while getting the most watched genre in the cluster.

Our algorithm is as follows:

1. Get User Id as input.
2. Get the cluster in which this user is allocated and get all the information in that cluster relating to all the users in the cluster.
3. Separate only the genres column from the rows retrieved.
4. Now, from the above list, we know that there are multiple genres in one that is separated with ‘|’. Hence, we split the string and got the genres separately. So, for each genre we increased the count. This was repeated till we had counted all the genres in the given list.
5. Finally, we sorted the genres according to the count and extracted the topmost five.

The topmost five that we get over here is the generalized topmost five genres for that cluster. Hence, in this way, we could achieve the required goal for our project. The same algorithm is applied for the remaining methods as well.

One example of the result is as follows:

Enter the user ID of person: 9

Drama, Comedy, Action, Thriller, Adventure

**Step 4: Compare it with individual most watched genre**

Now that we have achieved the goal of our project, we should prove its validity. So, to prove the validity of the clusters, we came up with an idea. We are predicting individual top most preferred movie genre. So, if we could list out the top five most preferred genres for that individual only, then we could compare it with the result that we got in the previous step and get the percentage match which would prove the algorithm’s validity.

For this step, we got the individual user’s preference using the cluster itself. We know that each user’s information will be in that cluster only. So, if we apply similar algorithm as the previous step but the change would be that the algorithm will only consider the user’s information and get the five-topmost preference for the user. This would be an accurate one as this only considers the user’s information and has nothing to do with prediction. It will consider the facts only.

This was the final step of our project. So, the results from this project flow is discussed in the later topic.

1. *Result*

We used the same algorithm and process for two different number of clusters. That would be 35 and 50 as we already showed in the cluster diagram in above topic. So, to show our results clearly, we used visualization as well as statistical information. Firstly, this is how our project gives the result. Following shows the list of 5 most popular genre for user Id 123 and followed by that is the user’s individual preferences which is used to check the validity of the project.

Enter the User ID of the person: 9

Drama, Comedy, Action, Thriller, Adventure

Data Based on the individual person:

Drama, Romance, Comedy, Thriller, Crime

As we can see that there are three genres that are common in both the list i.e. Drama, Comedy and Thriller. So, it gave us a 60% match. We can say, with this result, that the results were satisfactory. To describe more about the clusters, we analyzed each cluster by checking how much percentage of user were in each cluster.

Following is the table that has information about the clusters, how many users are in that cluster and finally the percentage of users in that cluster. It has also been shown using a bar plot as follows.

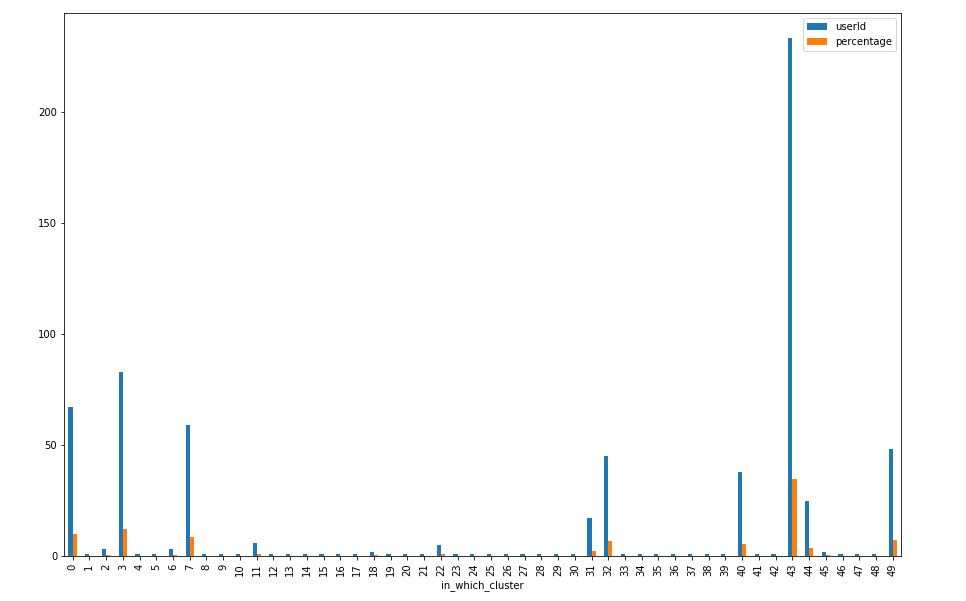
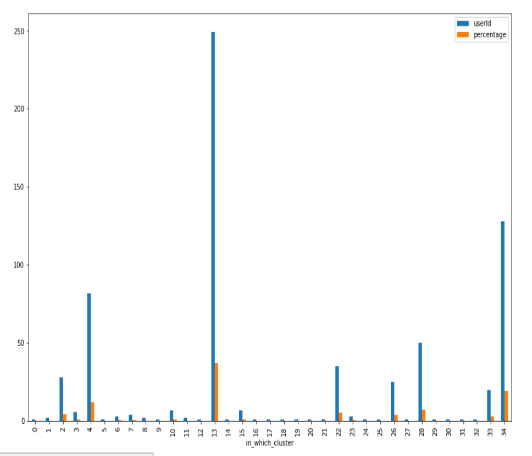
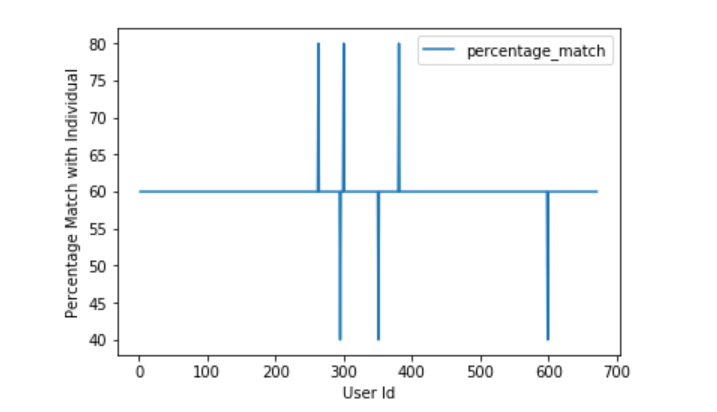
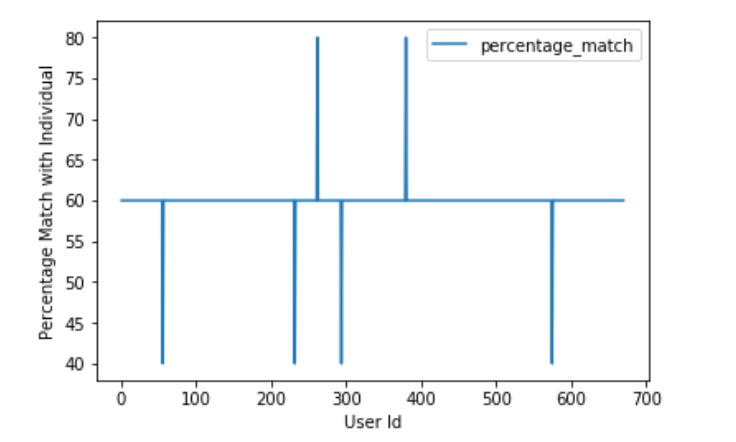


Fig. 14 Bar plot of percentage of users in a cluster and statistical information for 35 and 50 number of clusters

Finally, now we know that the results that we are getting are satisfactory. But how can we compare the validity of the clustering technique as a whole. For that, we ran the step 4 discussed in the previous topic for all the users in the dataset and got the percentage match for each one of them. Following is the result plotted as a line plot. In this most of them had 60% match, some of them had 40% match and some even had 80% match. As the x-axis has lesser frequency range, the line plot is coming that way but the lines that we see are actually representing a lot of users.



Fig. 15 Percentage accuracy for 35(top) and 50(bottom) clusters

From this above plot, we can say that as there are more number of 80% matches when we are using K-Means Clustering, 35 would be an ideal choice for optimum number of clusters for this project.

1. Comparative Methods

For the comparative analysis, we have used Spectral Clustering and Mean-Shift. The description, program flow and the experimental results for the two methods are discussed in detail below followed by their comparison with K-Means Clustering.

## Spectral Clustering

1. *Description*

Spectral clustering is a family of methods to find K clusters using the eigenvectors of a matrix. It first constructs the affinity matrix which is then represented as EigenVectors using PCA and finally the eigenvalues related to those eigen vectors are used to plot the points in the clusters.

1. *Project Flow*

Now, the project flow is actually similar to the K-Means itself and we have used the same number of clusters for the experimentation i.e. 35 and 50. The only difference is that we have used the built-in Spectral Clustering function and the results are drastically different.

We just need to input the number of clusters and the correlation table that we have made to the Spectral clustering function to get the clusters.



1. *Result*

Similar to the K-Means Clustering, we got two different cluster arrangements for 35 clusters and 50 clusters. And the results were similar for both 50 as well as 35 clusters. Following was the resulting scatter plot.

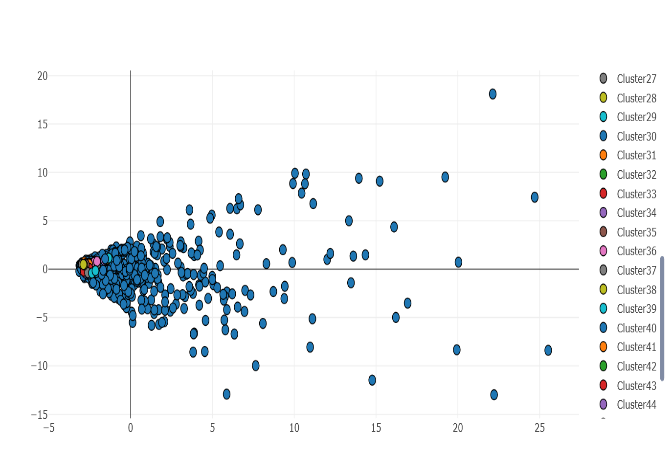
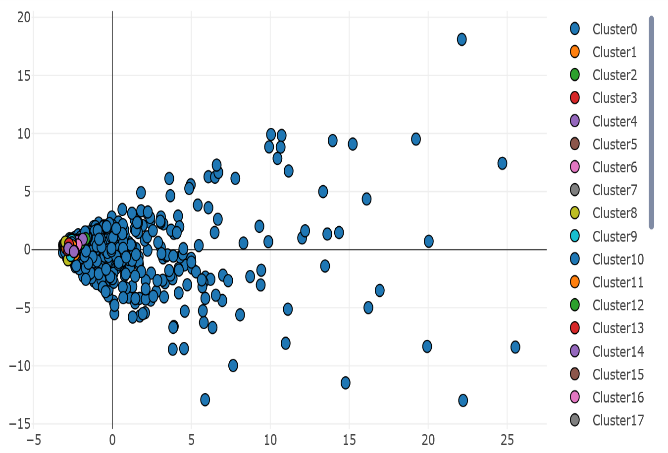
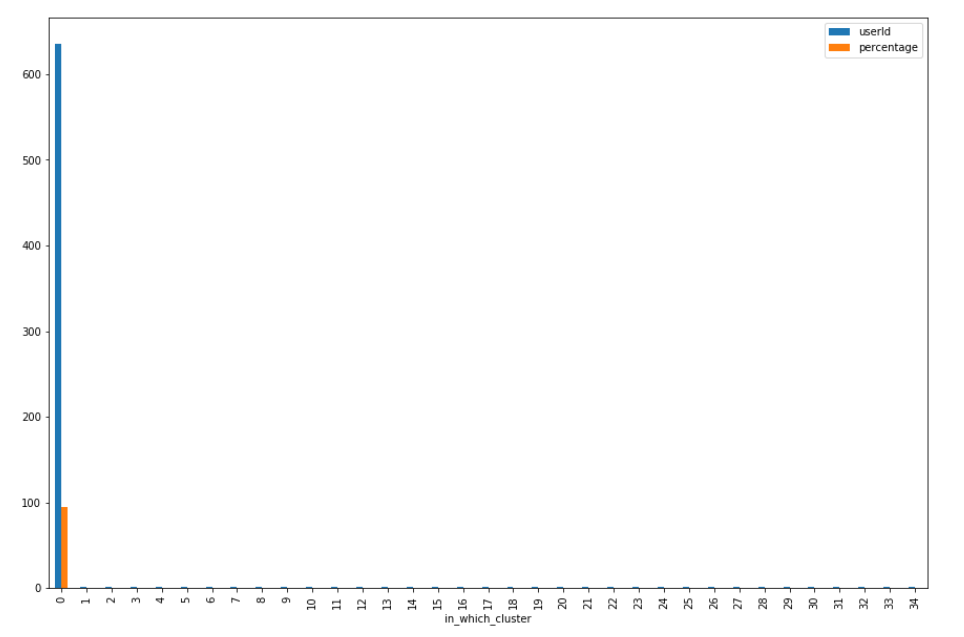


Fig. 16 Spectral Clustering using 35(top) and 50(bottom) clusters

And after running our algorithm, we got the prediction similar to K-Means. We get similar output like the ones in K-Means.

Now, following was the percentage of user in each cluster. Over here, we can actually see that most of the users are in concentrated in one cluster and rest of the clusters have a few users in it. But we still cannot judge the technique just because of this.



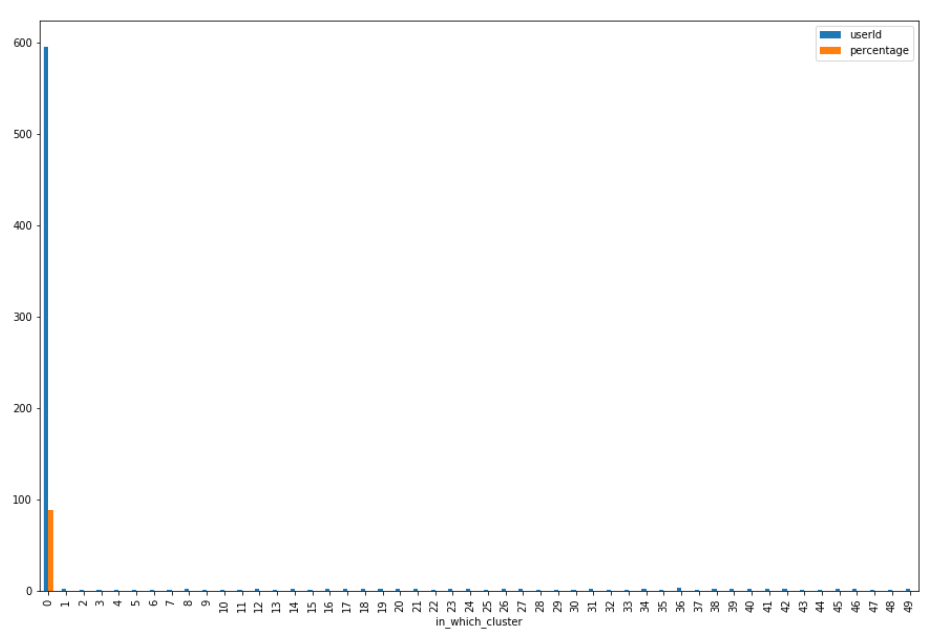


Fig. 17 Bar plot of percentage of users in a cluster and statistical information for 35(left) and 50(right) number of clusters

So, to check the validation of this technique, we applied the same step 4 for each user and got the percentage match and it came as follows.

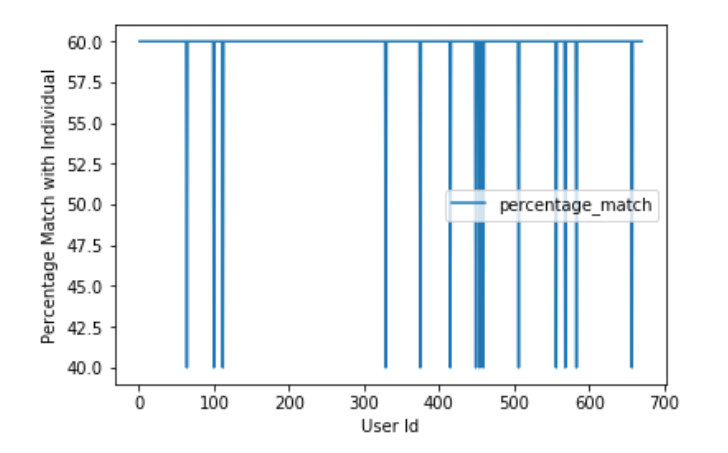
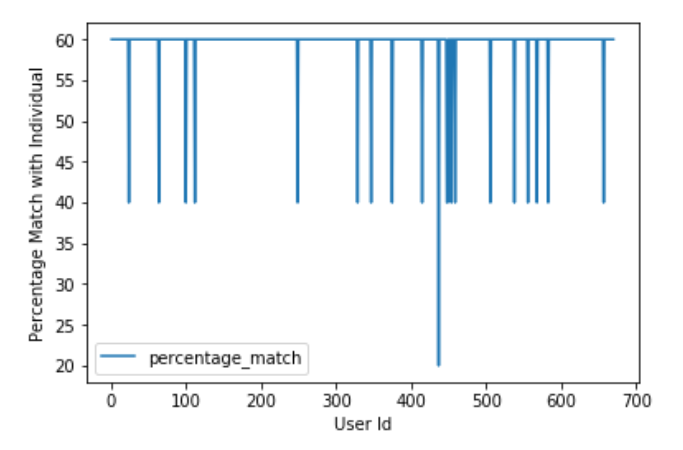


Fig. 18 Percentage accuracy for 35(top) and 50(bottom) clusters

From this result, we can see that 35 cluster is much better than 50 clusters as while using 35 clusters the least accuracy that it gave was 40% but while using 50 clusters, the least accuracy it achieved was 20%. So, using 35 clusters for Spectral Clustering was found to be better for this project.

But when we compare it with K-Means Clustering, we can see that K-Means clustering is far better than Spectral Clustering looking at the highest accuracy reached. The reason for this would be discussed in later topics.

## Mean-Shift Clustering

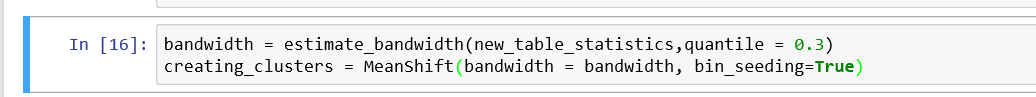
1. *Description*

Mean shift is a hill climbing algorithm which involves shifting of kernel which is radius r of a circular window centered at a point, iteratively to a higher density region until the algorithm converges. Mean shift vector defines every shift towards the direction of the maximum increase in the density. The kernel is shifted to the centroid or the mean of the points within it at every iteration. The new centroid is selected by calculating the mean of the points encircled by the radius of the circular window.

1. *Project Flow*

Similar to the previous two methods, the implementation goes the same way till the PCA gets two different PCA components. After that the only difference is we are using Mean Shift clustering rather than the K-Means clustering. Sklearn has a built-in library for Mean Shift Clustering as well. Once the clusters form, the rest of the algorithm are pretty much the same.

For Mean-shift clustering, we have to first specify the bandwidth via which the clustering start its process. For the bandwidth, we have fed it with the correlation table and quantile value as 0.3. All the rest of the values that have been used are default values.



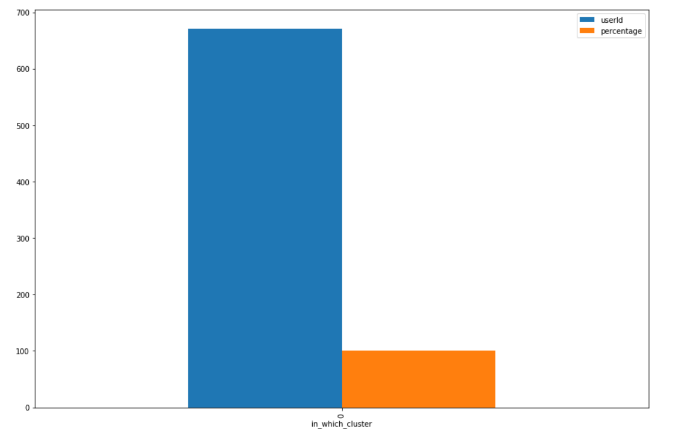
1. *Result*

The result that we got for this clustering technique is more accurate than what we got for spectral clustering even though only one cluster formed at the end. Result was similar to the one that was given in the previous techniques.

User Id: 123

List: Drama, Comedy, Action, Thriller, Adventure

As only one cluster got formed, all the users got clustered into one cluster and so when we plotted that visually just like we did to all other techniques, Figure 19 was the bar plot that we got.



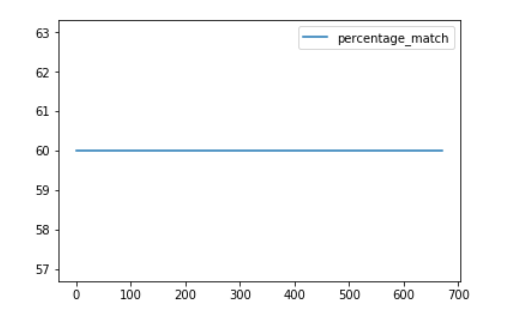


Fig. 19 Bar plot for showing percentage distribution of user across clusters and mean-shift prediction accuracy

Now, we know that we shouldn’t just rely on the visual observation. So, just like for the other techniques, we applied our prediction accuracy techniques. And Figure 19 prediction accuracy was the result.

Even though it only formed one cluster, it is still giving a better prediction for this particular dataset than spectral clustering as the percentage of match is 60% and it remained constant throughout the entire dataset.

1. Project analysis

To check the validity of the project in itself, we applied our own algorithm as we stated in the previous topics. Still in short, after clusters are formed, we selected one cluster and got the generalized preference from that cluster. Then, we selected individual user from that particular cluster and got his preferences and we compared both results to estimate the accuracy of our algorithm.

All the clustering methods gave the satisfactory accuracy as mentioned in above. K-Means gave 40-80% accuracy, Spectral gave 20-60% accuracy and finally, Mean-Shift gave a constant 60% accuracy. So, in a whole, we got satisfactory results for all the techniques that we implemented. The detailed discussion of how each of the techniques are in the following topic.

1. Comparative analysis

We are comparing the clusters by using visualization.

1. K-Means Clustering

We are achieving upto 80% accuracy and a least of 40% accuracy so it is a pretty good clustering technique for our project from Figure 15. The reason why this happened is described in the following paragraph.

In K-Means Clustering, it was able to form good clusters because of the fact that the k means is based on distance calculated between the point and the centroid. The data point is included into a cluster if its distance from centroid is minimum. Even though the data points of our dataset are close to each other, we are still able to form good clusters in comparison to the other techniques as the clustering doesn’t depend on the distance between the two points rather it’s distance to the centroid.

1. Spectral Clustering

We can see from the prediction accuracy that it is only attaining to highest of 60% accuracy and least to 20% as well from Figure 18. So, this is actually not a good clustering technique for this project. The reason behind this is discussed in detail in the following paragraph.

In Spectral Clustering, the clusters are formed based on the graph connectivity of the points. It forms Affinity matrix between points and based on the similarity, the clusters are formed. In our dataset, all data points are compact and similar to each other and connected thus forming few clusters and majority of points are in one big cluster. And only a few points that differ from each other. So, it was not forming good clusters.

1. Mean Shift

We can see that the highest as well as the lowest accuracy of Mean Shift is 60% from Figure 19. So, we can say that as a whole, this proves to be better than Spectral Clustering but still not as good as K-Means Clustering. The reason behind this is discussed later in this topic.

In Mean shift, the clusters are formed based on cluster head and the bandwidth or pilot window. Now, it calculates the mean of every point in that window and the point with maximum mean will be considered as next cluster head. This is repeated until there is no change in cluster heads. In our dataset, while performing mean shift, the data points are kept on adding to single cluster head as the dataset is compact i.e. the points are more similar than dissimilar and thus forming only one big cluster at the end. Thus, it was not able to find good clusters with our dataset.

Thus, by comparing all the clustering methods, K-Means is the best clustering method worked in our project.

1. Future work

In future, we want to further improve our project by including the individual movie’s ratings in our prediction, so that we improve our recommendations by recommending only those movies from his/her top preferred genres which are rated best. This way, the user will be able to get recommendation not only from his/her preferred genre, but also the movies which are rated best in that genre by others.

Also, we want to validate our accuracy prediction by using other techniques that are more effective and can be depended on. Finally, as we are using sample dataset in our project, it won't remain same always, so we want to test our project with actual dataset in future implementation, by this we can check the validity of algorithm with more datasets and then get actual accuracy of our technique.

1. Conclusion

In conclusion, we can say that we have learnt a lot from this project about the three-different clustering technique that we implemented i.e. K-Means, Spectral and Mean-Shift Clustering. We learnt what kind of datasets it works on and it doesn’t. We specifically learnt how the compactness of a data affect the accuracy of clustering of different techniques.

For this project in specific, we got that the spectral and mean shift clustering were not good with compact dataset and k-means proved to best with our dataset. And K-Means worked best with 35 clusters. We got 80% as our highest accuracy and lowest as 40%. So, we know that we can depend on this type of technique.

This project helped us a lot in understanding the data mining techniques in implementation rather than in theory.

1. References

<https://ieeexplore.ieee.org/abstract/document/6784640/>

<https://ieeexplore.ieee.org/abstract/document/6779375/>

<https://grouplens.org/datasets/movielens/>

<http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

<http://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html>

<http://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html>

<http://scikit-learn.org/stable/>

<http://www.uokufa.edu.iq/staff/ehsanali/Tan.pdf>

[Introduction to Data Mining Pang Ning Tan Vipin Kumar Text book.](http://www.uokufa.edu.iq/staff/ehsanali/Tan.pdf)

<https://www.python.org/>