COMP 5411 FA
Fall 2019
Final Project Report

Movie Recommendation System using Content-Based Filtering

Group Members:

Name	Student No.	Section
Kunal Naresh Kumar Thapar	1107686	COMP 5411 FA
Rahul Niraj Singh	1099198	COMP 5411 FA
Charlie W. Dondapati	1111674	COMP 5411 FA

Abstract

Background

A recommendation engine is an intelligent system that can provide suggestions to the users on resources like movies, songs, books or online inventory. These recommender systems have become a key module in many of the marketing and entertainment websites to draw more users, by suggesting content that aligns more with the choices of the user. Recommender systems predict user choices based on the user's history and inputs.

Objective

This project is aimed at developing a recommendation system based on content-based filtering that suggests new movies to the user, analyzing the ratings of the previously watched films and user preferences.

Methods

Based on the similarity of the genres the movies are grouped in clusters. K-means clustering algorithm is implemented on the movies dataset to establish clusters of similar movies genres. Furthermore, analyzing the ratings of a specific user which he/she has given to other movies, the average rating is calculated for each cluster and the top 10 films from the cluster where the maximum average rating is obtained are suggested to the user.The top 10 movies are recommended based on the ratings of other users.

Results

After implementing the K-means clustering algorithm, top-10 films similar to the ones in which the user has shown interest by rating the most are suggested to watch, predicting the user's choice.

Conclusion

A novel recommender system is developed based on content-based filtering to suggest new movies to the user depending on the ratings the user has previously provided. All the movies the user has rated are associated with the clusters of similar genres and the movies from top-rated genres by the user are recommended.

Introduction

The recommendation systems have gained popularity in many areas of technology. In recent times, many organizations have adopted this methodology to provide suggestions to better service the target users online. Some of the organizations which use this kind of technology are Amazon, LinkedIn, YouTube, Netflix and so on. Because the amount of data on the internet is ever-growing and the number of options available for the users is humungous, it becomes a time-consuming process for the user to select the products of their own taste. With the help of recommendation systems, both users and the organizations benefit equally by prompting the user with relevant choices and adding satisfying customers to the organizations. Movie Recommendation System proposed in this project uses Content-Based Filtering to suggest mew movies to the user, based on his/her historical data, such as genre and analyzing the details such as frequently accessed or liked pages.

A movie recommendation is important in our social life due to its strength in providing enhanced entertainment. Netflix is one of the pioneers in popularizing movie recommender systems by conducting

Netflix prize competition. Such a system can suggest a set of movies to users based on their interests and ratings. However, the Recommender System suffers from many challenges such as Lack of Data, Changing Data, Changing User Preferences, Unpredictable Items, Scalability, Privacy protection. To overcome these challenges, we implemented the K-Means Clustering method.

Literature review

The recommender system is used so extensively these days that it has become a preferable choice for researchers. The first paper on the recommender system was published in the year 1998[1]. Since then a significant number of papers had been published. Different factors have been explained to increase the reliability of the recommender system. In the year 2005 John O'Donovan[2], Barry Smyth has taken trust as the percentage of correct predictions that a profile has made in general (profile-level trust) or with respect to an item (item-level trust).

Another model on the content-based recommendation system has been proposed by Balabanovic et al. [3]. It could be applied in many different domains, such as books, movies, videos, or music. Generally, TF-IDF and Information Gain (IG) are used to analyze features like author, genre, and most frequently used words.The referenced papers used genre and rating matrix for determining the similarity and also the cosine similarity matrix (Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity).

In this era of the web, we have a huge amount of information overloading the Internet. It becomes a herculean task for the user to get the relevant information. To some extent, the problem is being solved by the search engines, but they do not provide the personalization of data. So, to further filter the information, we need a recommendation engine.

Data

The dataset used in our project is obtained from MovieLens 20M. MovieLens is one of the featured projects of GroupLens research lab at the University of Minnesota. GroupLens Research has collected and made available rating data sets from the MovieLens web site. MovieLens is a web site that helps people find movies to watch. The data sets were collected over various periods of time, depending on the size of the set. It has hundreds of thousands of registered users[4]. MovieLens conducts online field experiments in the areas of automated content recommendation, recommendation interfaces, tagging-based recommenders and interfaces, member-maintained databases, and intelligent user interface design.

Description

The dataset contains 10, 48,576 ratings for 58,098 movies from 10,532 users. Each user has more than 20 ratings. The ratings for each movie are ranked from 1 to 5 with 5 being the highest rating and 1 being the lowest rating. The dataset can be accessed from the given below-given link:

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

The dataset is classified into two parts as a movie dataset and the ratings dataset. The movie dataset contains the details related to movies and each movie can be associated with multiple genres. The ratings dataset consists of user details and the inputs provided by the user, such as rating for the previously watched movies.

Features:

Each dataset consists of three features which are explained in the table.

Movie dataset		
Attribute Name	**Attribute Type**	**Attribute description**
Movie Id	Continuous Data	Unique id for each movie
Title	Text Data	Movie title
Genres	Categorical Data	Associated genre for each movie
Ratings dataset		
Attribute Name	**Attribute Type**	**Attribute description**
User Id	Continuous Data	Unique id for users
Movie Id	Continuous Data	Refernce id from movie dataset
ratings	Numerical Data	ratings given by the user for the movies

Instances

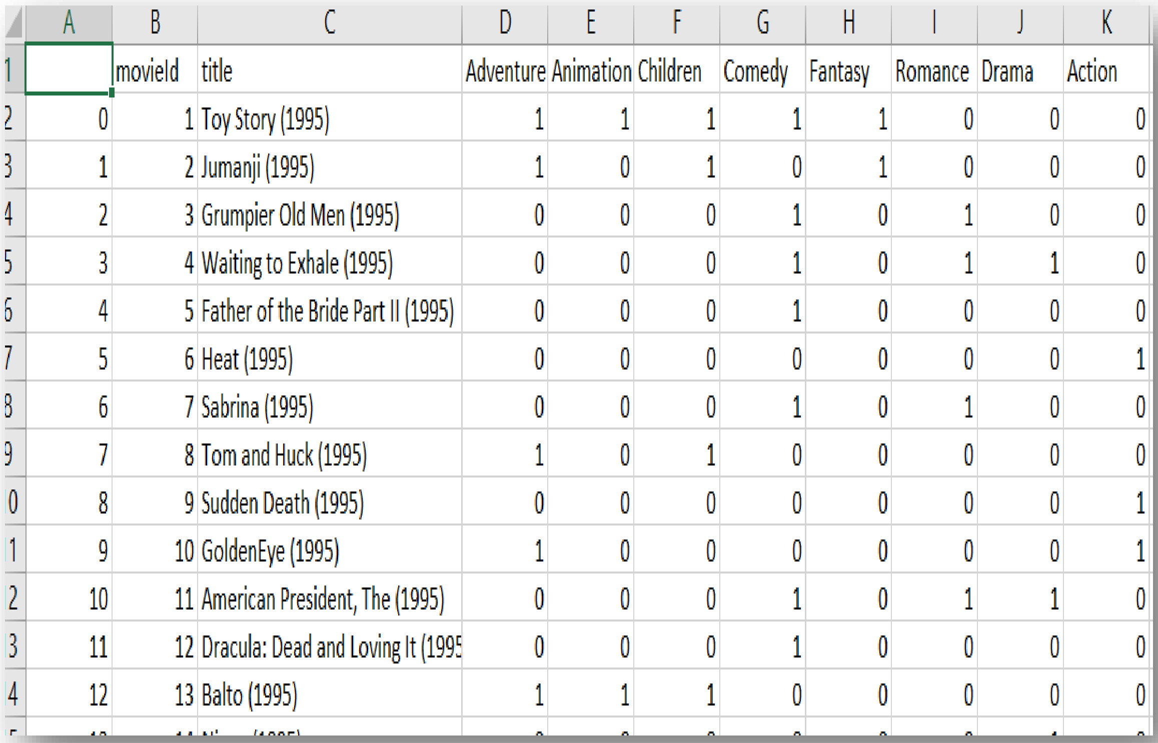
There is a total of 10, 48,576 ratings for 58,098 movies and there are 4,266 missing values

Methods and tools

Data preprocessing

Data analysis involves crucial steps of giving structure to the skewed data. One of the steps necessary for cleaning the data is dealing with the missing values. The missing values in the dataset are deleted before proceeding with clustering the reason is, On the basis of only movie id and title, it is not possible to predict the film genre.

Furthermore, each movie is associated with more than one genre. The Genres feature is converted to multiple features of categorical numerical values 0 and 1 as shown below. The value 1 represents the presence and 0 represents the absence of the respective genre.



	A	B	C	D	E	F	G	H	I	J	K
1		movieId	title	Adventure	Animation	Children	Comedy	Fantasy	Romance	Drama	Action
2	0	1	Toy Story (1995)	1	1	1	1	1	0	0	0
3	1	2	Jumanji (1995)	1	0	1	0	1	0	0	0
4	2	3	Grumpier Old Men (1995)	0	0	0	1	0	1	0	0
5	3	4	Waiting to Exhale (1995)	0	0	0	1	0	1	1	0
6	4	5	Father of the Bride Part II (1995)	0	0	0	1	0	0	0	0
7	5	6	Heat (1995)	0	0	0	0	0	0	0	1
8	6	7	Sabrina (1995)	0	0	0	1	0	1	0	0
9	7	8	Tom and Huck (1995)	1	0	1	0	0	0	0	0
0	8	9	Sudden Death (1995)	0	0	0	0	0	0	0	1
1	9	10	GoldenEye (1995)	1	0	0	0	0	0	0	1
2	10	11	American President, The (1995)	0	0	0	1	0	1	1	0
3	11	12	Dracula: Dead and Loving It (1995	0	0	0	1	0	0	0	0
4	12	13	Balto (1995)	1	1	1	0	0	0	0	0

Learning methods

As discussed previously, the content-based filtering approach is used for building a recommendation engine. Based on the data provided by the user, a user profile is generated which is then used to make suggestions to the user. Content-based filtering uses user's inputs to recommend choices similar to the user's choice of liking. As the user provides more inputs or takes action on the recommendations, the engine becomes more and more accurate. This recommendation model aims at suggesting movies based on the similarity of genres and the ratings of the users. If a user gives a high rating for a certain type of movie, a cluster of movies is identified which groups movies of similar genres and is recommended to the user. The movies suggested are based on the top ratings of that cluster.

In this project, K-means clustering is implemented to cluster similar movies based on associated genres. K-means clustering is an unsupervised method that is defined by targeted number k, which refers to the number of centroids you need in the dataset. A centroid is the imaginary or real location representing the center of the cluster. Every data point is allocated to each of the clusters by reducing the in-cluster sum of squares.

The sum of squared distances is calculated using the Euclidean distance. For any given to data points, the Euclidean distance is calculated by taking the square root of the sum of squares of the differences between the data points. For two data points with two attributes, the below formula is used to calculate the distance.

$$\sqrt{(x2-x1)2+(y2-y1)2}.$$

As the dataset used in this project is classified into 2 different sections, with one section containing the list of movies along with associated genres and the other containing a list of user ratings for the movies rated on a scale of 1–5, with 5 being the highest. First, the dataset has to be preprocessed for removing missing values and the genres feature is converted to multiple columns with binary values indicating the presence or absence of respective categories. Next, a combined relational dataset of movies with genres and user ratings has to be constructed for correlating genres with the ratings. Next, K-means clustering is used to cluster movies with similarity in genres and the sum of squared error (SSE is the sum of the squared differences between each observation and its centroid. It can be used as a measure of variation within a cluster. If all cases within a cluster are identical the SSE would then be equal to 0) is calculated. And it is followed by taking the average rating for the user from the correlated ratings dataset, with cluster value, and the top 10 movies from the respective cluster are presented to the user.If a new user is present and has not provided any rating then random movies will be suggested to them. The algorithm developed for building the recommendation model is given below:

Algorithm

Step1: Preprocess data to remove missing values

Step2: Preprocess data to convert genres feature into multiple columns with binary values 0 or 1. If a genre is present in a movie, a value of 1 is assigned or 0 is assigned

Step 3: Implement K-means clustering with initial values k =2 and 3

Step 4: Calculate the sum of squared error for the clusters and check with the decision based on sum of squared error value set at 3%

Step 5: If the sum of squared error is greater than decision values then increment k and repeat step 3 or go to step 6

Step 6: Correlate cluster values with ratings dataset and generate the requested user profile.

Step 7: For the given user, group by cluster and take an average of ratings provided.

Step 8: Return the top 10 movies from the cluster where you get maximum average excluding the ones which has already been rated by the user

Validation methods

1. To validate the algorithm we have calculated silhouette score for different values of K based on the decision criteria. **Silhoutte Score:** The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters

	Decision less than 3%	Decision less than 2%	Decision less than 1.5%	Decision less than 0.1%
Value of K	10	16	24	28
Silhouette Score	0.378	0.456	0.52	0.539

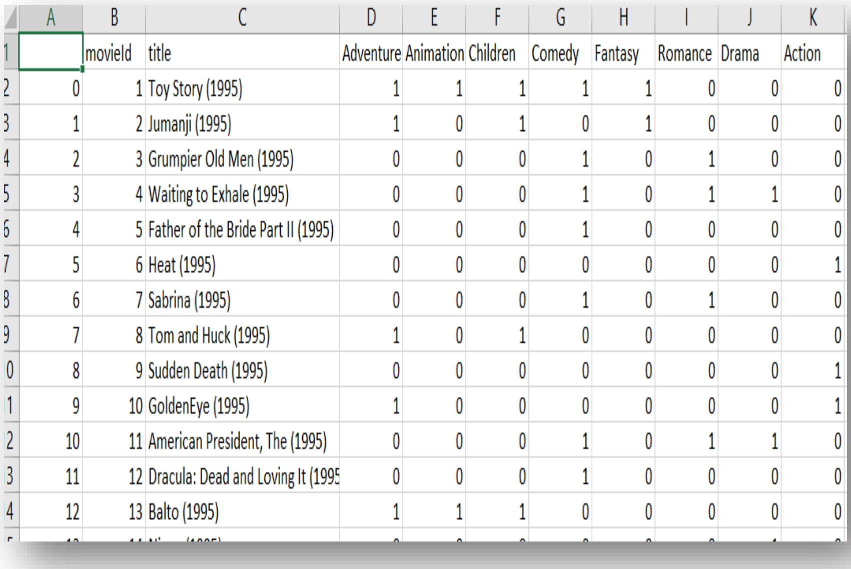
2. If a new user is present and has not provided any rating then random movies will be suggested to them.

Tools

- Programming language : Python
- Version : 3.7
- IDE : Spyder
- Libraries : Numpy, sk-learn, pandas

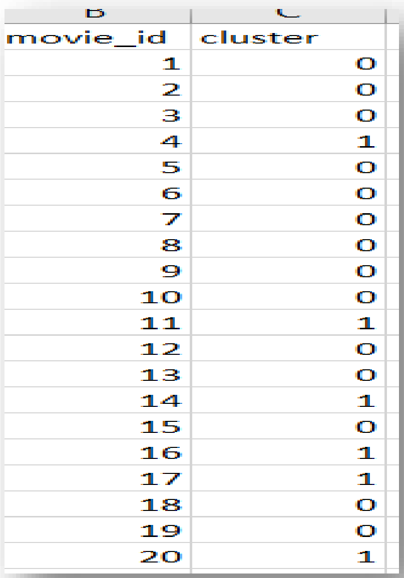
Results

The genres feature is separated into multiple columns with the binary values 0 and 1 indicating whether the movie belongs to the designated genre or not. The value 1 represents the association of the genre with the movie and the value 0 represents otherwise.



A	B	C	D	E	F	G	H	I	J	K
	movieId	title	Adventure	Animation	Children	Comedy	Fantasy	Romance	Drama	Action
0	1	Toy Story (1995)	1	1	1	1	1	0	0	0
1	2	Jumanji (1995)	1	0	1	0	1	0	0	0
2	3	Grumpier Old Men (1995)	0	0	0	1	0	1	0	0
3	4	Waiting to Exhale (1995)	0	0	0	1	0	1	1	0
4	5	Father of the Bride Part II (1995)	0	0	0	1	0	0	0	0
5	6	Heat (1995)	0	0	0	0	0	0	0	1
6	7	Sabrina (1995)	0	0	0	1	0	1	0	0
7	8	Tom and Huck (1995)	1	0	1	0	0	0	0	0
8	9	Sudden Death (1995)	0	0	0	0	0	0	0	1
9	10	GoldenEye (1995)	1	0	0	0	0	0	0	1
10	11	American President, The (1995)	0	0	0	1	0	1	1	0
11	12	Dracula: Dead and Loving It (1995	0	0	0	1	0	0	0	0
12	13	Balto (1995)	1	1	1	0	0	0	0	0

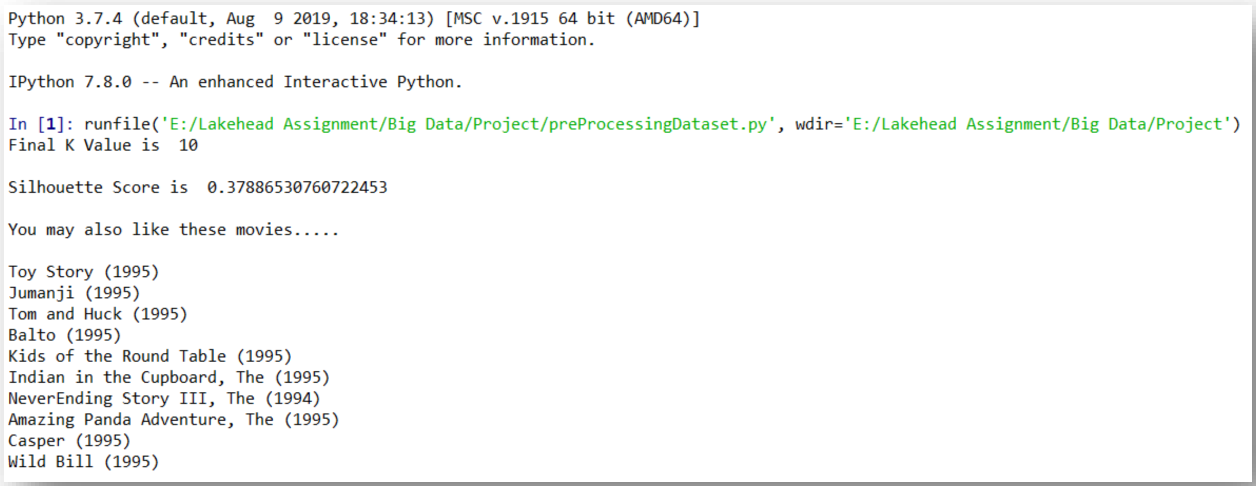
The clusters created for the movies based on the similarity of the genres are correlated with the ratings and movie dataset, with the respective clusters as shown below and thus creating a combined relational dataset.



movie_id	cluster
1	0
2	0
3	0
4	1
5	0
6	0
7	0
8	0
9	0
10	0
11	1
12	0
13	0
14	1
15	0
16	1
17	1
18	0
19	0
20	1

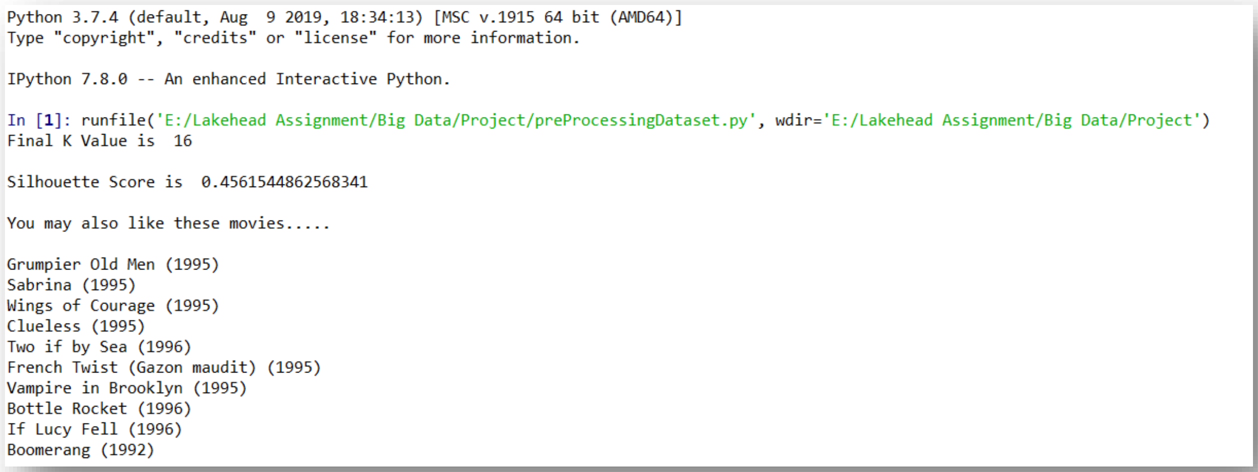
Based on the average rating for the given user, a cluster with the maximum average is determined and the top 10 movies are suggested to the user.

1. With SSE value 3% the results are as follows:

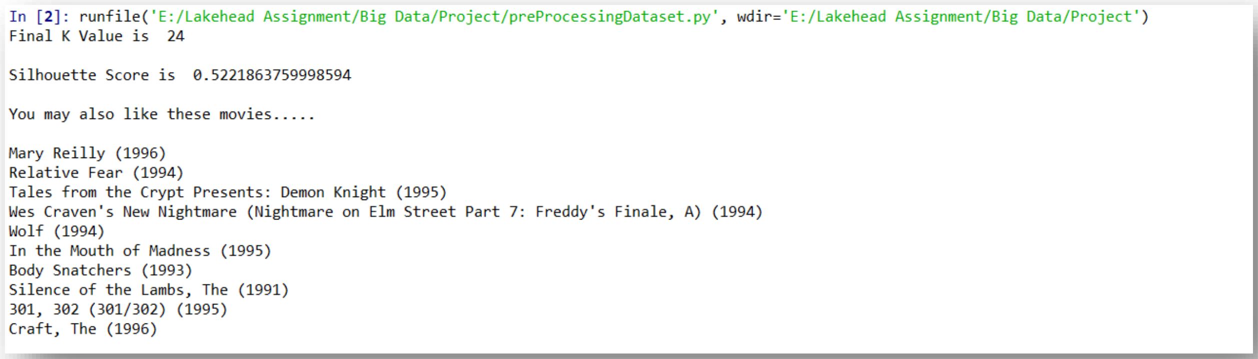


```
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.8.0 -- An enhanced Interactive Python.

In [1]: runfile('E:/Lakehead Assignment/Big Data/Project/preProcessingDataset.py', wdir='E:/Lakehead Assignment/Big Data/Project')
Final K Value is  10

Silhouette Score is  0.37886530760722453

You may also like these movies.....

Toy Story (1995)
Jumanji (1995)
Tom and Huck (1995)
Balto (1995)
Kids of the Round Table (1995)
Indian in the Cupboard, The (1995)
NeverEnding Story III, The (1994)
Amazing Panda Adventure, The (1995)
Casper (1995)
Wild Bill (1995)
```

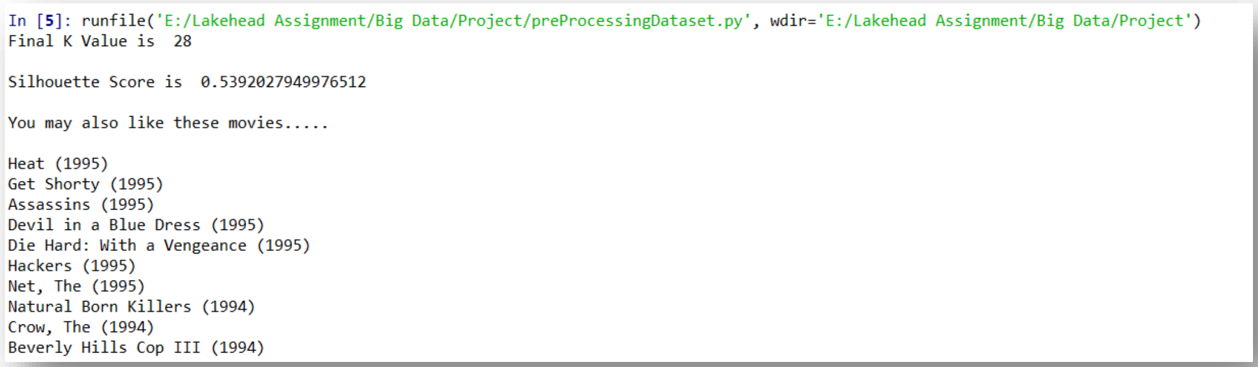
2. With SSE value 2% the results are as follows:


```
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.8.0 -- An enhanced Interactive Python.

In [1]: runfile('E:/Lakehead Assignment/Big Data/Project/preProcessingDataset.py', wdir='E:/Lakehead Assignment/Big Data/Project')
Final K Value is  16

Silhouette Score is  0.4561544862568341

You may also like these movies.....

Grumpier Old Men (1995)
Sabrina (1995)
Wings of Courage (1995)
Clueless (1995)
Two if by Sea (1996)
French Twist (Gazon maudit) (1995)
Vampire in Brooklyn (1995)
Bottle Rocket (1996)
If Lucy Fell (1996)
Boomerang (1992)
```

3. With SSE value 1.5% the results are as follows:


```
In [2]: runfile('E:/Lakehead Assignment/Big Data/Project/preProcessingDataset.py', wdir='E:/Lakehead Assignment/Big Data/Project')
Final K Value is  24

Silhouette Score is  0.5221863759998594

You may also like these movies.....

Mary Reilly (1996)
Relative Fear (1994)
Tales from the Crypt Presents: Demon Knight (1995)
Wes Craven's New Nightmare (Nightmare on Elm Street Part 7: Freddy's Finale, A) (1994)
Wolf (1994)
In the Mouth of Madness (1995)
Body Snatchers (1993)
Silence of the Lambs, The (1991)
301, 302 (301/302) (1995)
Craft, The (1996)
```

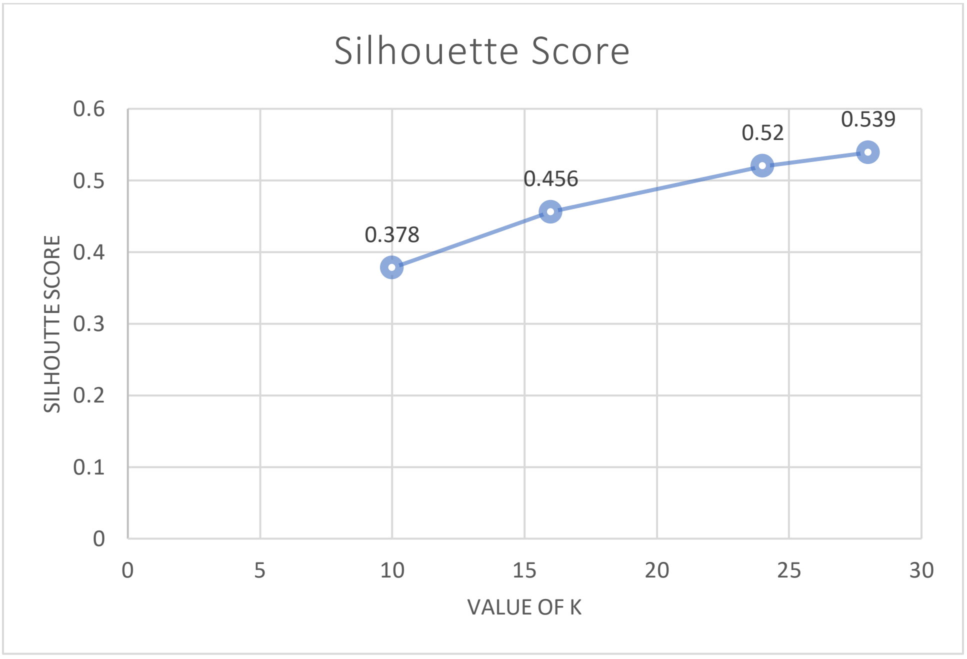
4. With SSE value 0.1% the results are as follows:


```
In [5]: runfile('E:/Lakehead Assignment/Big Data/Project/preProcessingDataset.py', wdir='E:/Lakehead Assignment/Big Data/Project')
Final K Value is  28

Silhouette Score is  0.5392027949976512

You may also like these movies.....

Heat (1995)
Get Shorty (1995)
Assassins (1995)
Devil in a Blue Dress (1995)
Die Hard: With a Vengeance (1995)
Hackers (1995)
Net, The (1995)
Natural Born Killers (1994)
Crow, The (1994)
Beverly Hills Cop III (1994)
```

Tabular Representation:

K Value	Silhouette Score
10	**0.378**
16	**0.456**
24	**0.52**
28	**0.539**

Graphical Representation:



Conclusion

The recommendation system developed in this project focuses on suggesting movie recommendations to the user, based on the genres of the movies and the user ratings provided for the previously watched movies. If a user gives a high rating for a movie that is associated with a genre, movies that belong to similar genres will be recommended to him. Recommendation systems are currently used widely in various avenues of today's digital technology. Many organizations are adopting recommendation engines to best serve the customer, predicting related items to ensure the better experience and ease of satisfaction targeting the user's previous choices. This recommendation model could be helpful, particularly, for video streaming companies or movie production houses to deliver reliable and relevant content. By employing machine learning methods in recommendation systems, intelligent recommendations can be made for customers and with the potential of such systems, these engines could add substantial value to both the organizations and the user.The limitation here is, If a new user is present and has not provided any rating then random movies will be suggested to them as no cluster will be created according to the genre.

References

[1] K. k. Parul, "Literature Survey: Recommender Systems," *International Journal of Engineering Technology, Management and Applied Sciences,* vol. Volume 3 , no. Special Issue, pp. 268-273, march 2010.

[2] B. S. John O'Donovan, "Trust in Recommender Systems," in *Proceedings of the 2005 International Conference on Intelligent User Interfaces*, San Diego, California, USA, January 2005.

[3] Y. S. Marko Balabanovic, "Fab: content-based, collaborative recommendation.(Special Section: Recommender Systems)," *Communications of the ACM,* vol. volume 40, no. 3, pp. p66-73, 1997.

[4] University of Minnesota, "GroupLens," [Online]. Available: https://grouplens.org/.

[5] S. Halder, A. J. Sarkar and Y.-K. Lee, "Movie Recommendation System Based on Movie Swarm," in *Second International Conference on Cloud and Green Computing*, Xiangtan, China, 2012.

[6] S. N. S. K. S. A. SRS Reddy, "Content-Based Movie Recommendation System Using Genre Correlation," Singapore, 2018.

Appendix

General Requirements:

1. Install sklearn using command $ `pip install scikit-learn==0.21.3`
2. Install numpy using command $ `pip install numpy==1.17.3`
3. Install numpy using command $ `pip install pandas==0.25.1`
4. The dataset files(movies.csv & ratings.csv) and the python code must be in the same folder
5. In the code go to line 269 which represents function **suggestFilms** and userid (the third parameter) can be changed manually from 5(default) to any number, Furthermore, the top suggestions count can be changed from 10 to any number of values which is the fourth parameter.
6. If there is a new user and has not provided any rating then random movies will be suggested to them.
7. In the code go to line 149 and change value 0.03(default) ie. 3% to alter the decision criteria based on SSE if need to check.
8. Run the python file on the console to execute the program