Machine Learning Project Report
Movie Recommend System

Group No.: ML19

Name	Id Number
Abhinav Talesra	2019AAPS0223H
Shatakshi Gupta	2019A3PS0397H
Parth Aggarwal	2019AAPS0218H
Akshat Sharma	2019A3PS0304H
Shradha Somani	2019A3PS0318H
Sahaj Gupta	2019A7PS0148H

Literature review:

1.A content-based movie recommender system based on temporal user preferences
In this paper, to dodge the utilization of content-based shifting, the Item-based CF shifting approach is utilized to get superior. KNN collaborative proposal framework is proposed utilizing cosine similitude by utilizing Movielens dataset containing 28M rating for over 60K motion pictures. The existing framework is compared, and found that the proposed framework is more solid and precise. It is additionally found that when the proposed strategy is connected to diverse bigger datasets, both exactness, and productivity increment which demonstrate that our framework is both exact and as well as effective. This item-based shifting is more convenient than user-based. The best point was to progress the normal proposal calculation and to supply way better things. The investigation work was fruitful because it has been able to fulfill our point of the venture. In the future, more highlights can be included to datasets (year of discharge, performing artist, class, casting points of interest, etc) to create suggestions more dependable and inventive. As a hybrid technique, content-based filtering and collaborative filtering can be coupled to reduce errors and enhance speed.

2. Movie Recommendation System Using Collaborative Filtering
In this paper, we have actualized a motion picture suggestion framework utilizing collaborative shifting. This framework is created utilizing Apache Mahout and takes the evaluations given to motion pictures into thought to supply motion picture recommendations. For future work, the recommender framework can be created utilizing a hybrid shifting approach rather than a collaborative. Later investigation demonstrates that hybrid frameworks are found to be more viable and give more exact suggestions. Hence, hybrid frameworks would be an enhancement. Our framework considers the client appraisals to suggest motion pictures. In the future, more features such as the class of the motion picture, the executives, the on-screen characters, and so on can be considered as well to supply proposals. In expansion, an unused system called Apache Expectation 10 might be looked into to create the framework rather than Mahout. The Apache Expectation 10 could be a machine learning server that employs the innovation stack of Apache Hadoop, Apache start, Flexible Look, and Apache Hbase to construct a Universal Recommender System.

3. Movie Recommender System Using Collaborative Filtering
The purpose of this paper is to provide an approach that increases the precision and performance of a standard filtering approach. Although there are other approaches for implementing a recommendation system, content-based filtering is the most straightforward. Which takes the user's input, double-checks his/her history/past behavior, and suggests a list of related films. To demonstrate the effectiveness, K-NN algorithms and collaborative filtering are utilized in this research, with the primary goal of improving the accuracy of outcomes over content-based filtering. This method is based on cosine similarity and employs the k-nearest neighbor algorithm

and collaborative filtering techniques while eliminating the shortcomings of content-based filtering. The use of Euclidean distance is recommended, but sine similarity is used because the accuracy of the sine angle and the equidistant distance of the film are about the same.

In this paper, to dodge the utilization of content-based shifting, the Item-based CF shifting approach is utilized for getting better outcomes. KNN collaborative proposal framework is proposed utilizing cosine closeness by utilizing Movielens dataset containing 28M rating for over 60K motion pictures. The existing framework is compared and found that the proposed framework is more dependable and exact. It is additionally found that when the proposed strategy is connected to diverse bigger datasets, both precision, and productivity increment which demonstrate that our framework is both exact and as well as productive. This item-based shifting is more convenient than user-based. The most point was to make strides in the normal proposal calculation. The inquiry about work was effective because it has been able to fulfill our point of the venture. In the future, more highlights can be included in datasets (year of discharge, on-screen character, class, casting subtle elements, etc) to form suggestions more dependable and inventive. The content-based filtering and collaborative filtering can be combined to play down the mistakes and move forward with the execution as a hybrid approach.

4. A Movie Recommender System: MOVREC using Machine Learning Techniques

In this paper, we have presented Motion picture REC, a recommender system for motion picture suggestions. It permits a client to choose his choices from a given set of qualities and after that prescribe him a movie list based on the aggregate weight of diverse traits and utilizing K-means calculation. By the nature of our framework, it is not a straightforward errand to assess the execution since there's no right or off-base proposal; it is fair to a matter of conclusions. Based on casual assessments that we carried out over a little set of clients we got a positive reaction from them. We would like to have a bigger information set that will empower more important results utilizing our framework. Also, we would like to incorporate diverse machine learning and clustering calculations and think about how the comparative comes about. Inevitably we would like to implement a web-based client interface that encompasses a client database and has the learning show custom-made to each client.

5. Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm

In the spread of data, how to rapidly observe one's favorite film in an enormous number of motion pictures becomes a vital issue. A customized proposal framework can assume a significant part particularly when the client has no reasonable objective film. In this paper, we plan and execute a film proposal framework model joined with the genuine necessities of film suggestion through investigating KNN calculation and cooperative shifting calculation. Then we give a definite guideline and engineering of the JavaEE framework social data set model. At last, the experimental outcomes showed that the framework has a decent proposal impact. Under the state of gigantic data, the prerequisites of film suggestion framework from film amature are expanding. This article plans and executes a total film proposal framework model in light of the KNN calculation, cooperative separating calculation and suggestion framework technology[10]. We give a nitty gritty plan and advancement cycle and test the security and high productivity of the trial framework through a proficient test. This paper has reference importance for the advancement of customized proposal innovation.

6. Movie Recommendation System Using Genome Tags and Content-Based Filtering
During the last decade, the importance of a recommendation system has increased dramatically. It is due to the fact that a good recommender system can assist people in their daily decision-making process. When it comes to movies, collaborative recommendation aims to help consumers by enlisting the help of other users who are similar to them or by recommending movies based on their shared history ratings. Genre is a popular meta tag for categorizing comparable films; however, because genres are binary in nature, they may not be the ideal approach to recommend. In this research, a hybrid strategy is suggested for recommending comparable movies based on movie genomic markers and content-based filtering. It reduces the number of duplicated and low-proportion-of-variance tags using principal component analysis (PCA) and Pearson correlation techniques, resulting in a reduction in computation complexity. Initial results show that genomic tags outperform existing methods in terms of finding comparable types of movies and providing more accurate and personalized recommendations.

Problem Statement

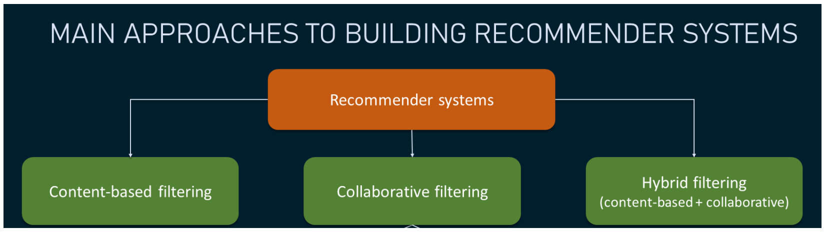
Movies are one of the sources of entertainment, but the problem is in finding the desired content from the ever-increasing millions of content every year. In this paper, to prove the effectiveness, K-NN algorithms and collaborative filtering are used to mainly focus on enhancing the accuracy of results as compared to content-based filtering.

Model Proposed in Research Paper

We are implementing the third paper from the literature review - Movie Recommender System Using Collaborative Filtering.
The model proposed uses an item based collaborative technique to recommend movies to the users. The item based approach is far more accurate and efficient to use as the item dataset is non-dynamic and can be made available offline whereas the user database keeps on changing. The proposed approach uses the KNN algorithm to find the distance between the target movies with every other movie in the dataset and then it ranks the top k nearest similar movies using cosine angle similarity.
Over this collaborative filtering is applied which further rank the movies based on the similarity with the movies liked before by the user.

KNN algorithm- is famous in a recommendation system for its faster predictive nature and low calculation time. KNN [16] classifies any unlabeled class to their respective classes by prediction on a similarity measure

Cosine similarity- to calculate the distance between the target movie and the movies in the dataset, cosine similarity is used. It measures the similarity between two documents irrespective of how different they are in size, and calculates the cosine angle between two vectors in multi-dimensional space.



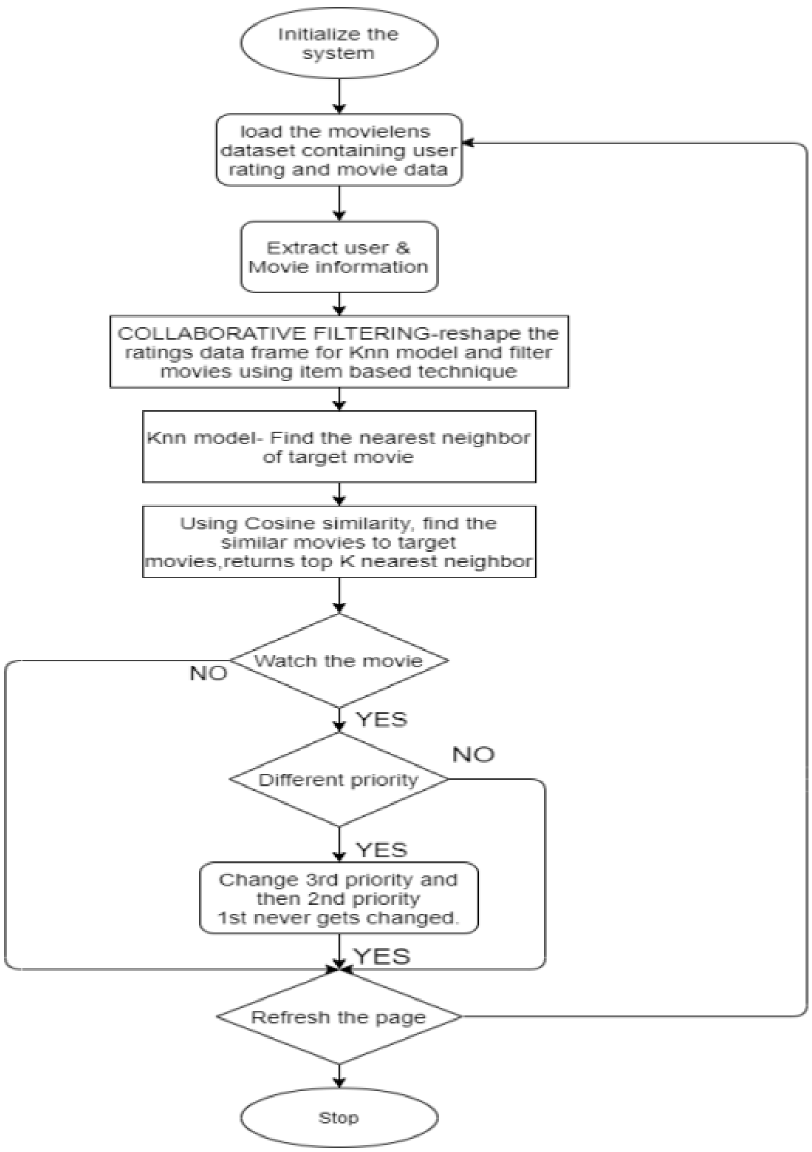


Figure 5: Proposed Collaborative filtering

Main code

```
reader = Reader()
```

```
ratings = pd.read_csv('../input/ratings_small.csv')
ratings.head()
```

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205

```
data = Dataset.load_from_df(ratings[['userId', 'movieId', 'rating']], reader)
data.split(n_folds=5)
```

```
svd = SVD()
evaluate(svd, data, measures=['RMSE', 'MAE'])
```

Innovation proposed

We proposed a simple hybrid recommender that brings together two techniques - content based and collaborative filter based engines.

Content based approach - This recommendation system requires some data or information on what the user might like or what his previous watched history. It is based on previous action or explicit feedback. Content based approaches are often unreliable when used alone because they require feedback data but used in hybrid mode, content based techniques can provide good results.

The hybrid approach-This approach provides very accurate results using both collaborative and content-based filtering while removing the drawbacks of the algorithms at the same time. This integrated system is getting more attention nowadays as it is better than both the algorithms

```
def convert_int(x):
    try:
        return int(x)
    except:
        return np.nan
```

```
id_map = pd.read_csv('../input/links_small.csv')[['movieId', 'tmdbId']]
id_map['tmdbId'] = id_map['tmdbId'].apply(convert_int)
id_map.columns = ['movieId', 'id']
id_map = id_map.merge(smd[['title', 'id']], on='id').set_index('title')
#id_map = id_map.set_index('tmdbId')
```

```
indices_map = id_map.set_index('id')
```

```
def hybrid(userId, title):
    idx = indices[title]
    tmdbId = id_map.loc[title]['id']
    #print(idx)
    movie_id = id_map.loc[title]['movieId']

    sim_scores = list(enumerate(cosine_sim[int(idx)]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:26]
    movie_indices = [i[0] for i in sim_scores]

    movies = smd.iloc[movie_indices][['title', 'vote_count', 'vote_average', 'year', 'id']]
    movies['est'] = movies['id'].apply(lambda x: svd.predict(userId, indices_map.loc[x]['movieI
d']).est)
    movies = movies.sort_values('est', ascending=False)
    return movies.head(10)
```

Results

Content Based Recommendation System Results:

In [26]:
```
get_recommendations('The Godfather').head(10)
```

Out[26]:
```
973      The Godfather: Part II
8387                 The Family
3509                       Made
4196         Johnny Dangerously
29               Shanghai Triad
5667                       Fury
2412             American Movie
1582    The Godfather: Part III
4221                    8 Women
2159             Summer of Sam
Name: title, dtype: object
```

In [27]:
```
get_recommendations('The Dark Knight').head(10)
```

Out[27]:
```
7931                      The Dark Knight Rises
132                              Batman Forever
1113                             Batman Returns
8227    Batman: The Dark Knight Returns, Part 2
7565                 Batman: Under the Red Hood
524                                      Batman
7901                           Batman: Year One
2579                Batman: Mask of the Phantasm
2696                                        JFK
8165    Batman: The Dark Knight Returns, Part 1
Name: title, dtype: object
```

In [53]:
```
improved_recommendations('The Dark Knight')
```

Out[53]:

	title	vote_count	vote_average	year	wr
7648	Inception	14075	8	2010	7.917588
8613	Interstellar	11187	8	2014	7.897107
6623	The Prestige	4510	8	2006	7.758148
3381	Memento	4168	8	2000	7.740175
8031	The Dark Knight Rises	9263	7	2012	6.921448
6218	Batman Begins	7511	7	2005	6.904127
1134	Batman Returns	1706	6	1992	5.846862
132	Batman Forever	1529	5	1995	5.054144
9024	Batman v Superman: Dawn of Justice	7189	5	2016	5.013943
1260	Batman & Robin	1447	4	1997	4.287233

In [54]:
```
improved_recommendations('Mean Girls')
```

Out[54]:

	title	vote_count	vote_average	year	wr
1547	The Breakfast Club	2189	7	1985	6.709602
390	Dazed and Confused	588	7	1993	6.254682
8883	The DUFF	1372	6	2015	5.818541
3712	The Princess Diaries	1063	6	2001	5.781086
4763	Freaky Friday	919	6	2003	5.757786
6277	Just Like Heaven	595	6	2005	5.681521
6959	The Spiderwick Chronicles	593	6	2008	5.680901
7494	American Pie Presents: The Book of Love	454	5	2009	5.119690
7332	Ghosts of Girlfriends Past	716	5	2009	5.092422
7905	Mr. Popper's Penguins	775	5	2011	5.087912

Collaborative Based Recommendation System Results:

```
In [67]: ratings[ratings['userId'] == 1]
```

Out[67]:

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205
5	1	1263	2.0	1260759151
6	1	1287	2.0	1260759187
7	1	1293	2.0	1260759148
8	1	1339	3.5	1260759125
9	1	1343	2.0	1260759131
10	1	1371	2.5	1260759135
11	1	1405	1.0	1260759203
12	1	1953	4.0	1260759191
13	1	2105	4.0	1260759139
14	1	2150	3.0	1260759194
15	1	2193	2.0	1260759198
16	1	2294	2.0	1260759108
17	1	2455	2.5	1260759113
18	1	2968	1.0	1260759200
19	1	3671	3.0	1260759117

```
In [68]: svd.predict(1, 302, 3)

Out[68]: Prediction(uid=1, iid=302, r_ui=3, est=2.673485594023968, details={'was_impossible': False})
```

Hybrid Based Recommendation System Results:

```
In [73]: hybrid(1, 'Avatar')
```

Out[73]:

	title	vote_count	vote_average	year	id	est
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	3.096906
974	Aliens	3282.0	7.7	1986	679	3.086270
8401	Star Trek Into Darkness	4479.0	7.4	2013	54138	3.060765
1011	The Terminator	4208.0	7.4	1984	218	3.012950
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	2.945010
2014	Fantastic Planet	140.0	7.6	1973	16306	2.881578
922	The Abyss	822.0	7.1	1989	2756	2.853512
1621	Darby O'Gill and the Little People	35.0	6.7	1959	18887	2.747493
7265	Dragonball Evolution	475.0	2.9	2009	14164	2.743694
1668	Return from Witch Mountain	38.0	5.6	1978	14822	2.730358

```
In [74]: hybrid(500, 'Avatar')
```

Out[74]:

	title	vote_count	vote_average	year	id	est
1011	The Terminator	4208.0	7.4	1984	218	3.435859
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	3.292873
2132	Superman II	642.0	6.5	1980	8536	3.160118
1621	Darby O'Gill and the Little People	35.0	6.7	1959	18887	3.141844
974	Aliens	3282.0	7.7	1986	679	3.139733
1668	Return from Witch Mountain	38.0	5.6	1978	14822	3.102818
8401	Star Trek Into Darkness	4479.0	7.4	2013	54138	2.987534
6084	Beastmaster 2: Through the Portal of Time	17.0	4.6	1991	27549	2.896088
8724	Jupiter Ascending	2816.0	5.2	2015	76757	2.896018
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	2.885750

Experiments conducted

We read and reviewed all the 6 papers. After analyzing all the papers, we implemented paper no. 3 in python. The approaches used were content based filtering and collaborative based filtering. In order to improve the model we implemented the hybrid approach which used both the above described techniques.

Main findings and accomplishments of your final project

Content Based Recommender: We built two content based engines; one that took movie overview and taglines as input and the other which took metadata such as cast, crew, genre and keywords to come up with predictions. We also devised a simple filter to give greater preference to movies with more votes and higher ratings.

Collaborative Filtering: We used the powerful Surprise Library to build a collaborative filter based on single value decomposition. The RMSE obtained was less than 1 and the engine gave estimated ratings for a given user and movie.

Hybrid Engine: We brought together ideas from content and collaborative filtering to build an engine that gave movie suggestions to a particular user based on the estimated ratings that it had internally calculated for that user.

Conclusion

We see that for our hybrid recommender, we get different recommendations for different users although the movie is the same. Hence, our recommendations are more personalized and tailored towards particular users. Compared to pure collaborative and content-based methods, hybrid methods can provide more accurate recommendations. They can also overcome the common issues in recommendation systems such as cold start and the data paucity troubles. We brought together ideas from content and collaborative filtering to build an engine that gave movie suggestions to a particular user based on the estimated ratings that it had internally calculated for that user.

Contribution

1. Abhinav Talesra : Collaborative Based Recommendation System, Report
2. Shatakshi Gupta : Hybrid Based Recommendation System, Report
3. Parth Aggarwal : Collaborative Based Recommendation System,Report
4. Akshat Sharma : Hybrid Based Recommendation System, Report
5. Shradha Somani : Content Based Recommendation System
6. Sahaj Gupta : Content Based Recommendation System, Report