ISSN No:-2456-2165

Songs Recommender System using Machine Learning Algorithm: SVD Algorithm

B.Srikanth[1], V.Nagalakshmi[2]
[1]PG Student, MCA, GIS, GITAM (Deemed to be University)
[2]Professor, GIS, GITAM (Deemed to be University)

Abstract:- Recommender systems have taken mainly on entertainment and e-commerce industries by storm. Variety of the samples of music recommender system is Amazon, Netflix, Spotify, Gaana Music. Music recommendation could also be a really challenging problem as we've to structure music during how that we recommend the favorite songs to clients which aren't a specific prediction. It's dynamic and sometimes leverage by components apart from clients' or songs' listening history. During this paper, I even have designed, implemented and analyzed songs recommendation system using SVD (Singular Value Decomposition) algorithm. According to .Nielsen's Music 360 2014 study, 93% of the U.S. population listens to music, spending quite 25 hours hebdomadally jamming bent their favorite tunes.

Keywords:- *Recommender system, collaborative filtering, Popularity Based, Content Based, Machine learning algorithm, SVD algorithm.*

I. INTRODUCTION

Music Recommender System could also be a system which learns from the user's past listening history and recommends those songs which they might likely to happen wish to listen to in future. I even have implemented SVD algorithm to make an efficient recommender system. Firstly implementing on popularity based model which was quite simple and intuitive. Most of the streaming services believe Collaborative filtering algorithms to suggest music. Collaborative filtering techniques which anticipate (filtering) taste of a client by gathering preferences and tastes from numerous different clients (collaborating) are also implemented. To beat this, top streaming services use a mixture of techniques to make what's called as a hybrid recommender system.

A. Popularity Based Model

Popularity Based is the foremost basic and straight forward technique. By discovering the recognition of every song by looking into the training set and calculating the amount of clients who had listened to the present song. Songs are then sorted with in the form of descending order according to their popularity. For every user, we recommend top hottest songs aside from those as of now in his profile. This technique includes no personalization and a couple of songs may never be listened in future.

B. Content Based Model

Content Based Model works with the info that user provides, either explicitly (rating) or Implicitly (Clicking on the link). Supported that data, a user profile is generated, which is employed to form suggestions to the user.

C. Collaborative Based Model

A collaborative filtering based model uses the prevailing history of the client and recommends music from other client's history which is analogous. For recommending, music is assessed according to the client's history, because the architectural complexity of the music is more, the efficiency of traditional classifiers reduces in classifying the music from different genres. Collaborative filtering uses the prevailing data to recommend the songs which reduces the complexity of the RS.

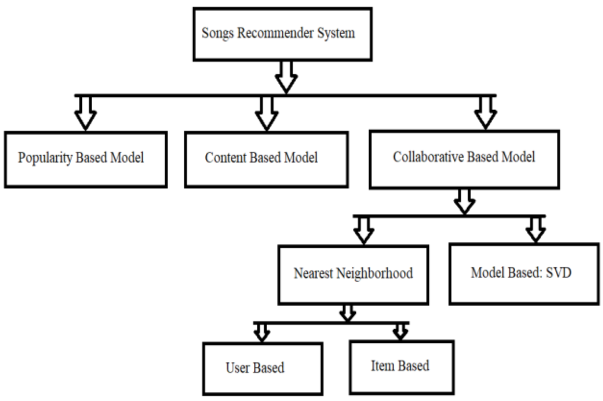

Fig 1:- Classification of various Recommendation Models

II. LITERATURE REVIEW

At present Recommender systems are using collaborative filtering techniques to gain an excellent success. Netflix giving an open challenge for the simplest collaborative filtering technique and therefore the winning technique uses latent factor models could make 10.09% improvements over the algorithm employed by Netflix at that point. Amazon uses user-user based and item-item based collaborative filtering which greatly contributes to the Songs Recommender System. Songs Recommender system shares some similarities with other commercial recommendation systems, but it focuses more on providing good and personalized advice on music, instead of goods for users to shop for. The perfect music Recommender system should be ready to automatically recommend personalized music to

ISSN No:-2456-2165

human listeners. Different from books or movies, the length of a bit of music is far shorter, and therefore the times that listening their favorite songs are commonly quite once, which are the most challenges we are getting to face.

III. MACHINE LEARNING ALGORITHMS: SVD ALGORITHM

A. SVD (Singular Value Decomposition) Algorithm

SVD could also be a matrix factorization technique that is typically wont to diminish the quantity of feature of a dataset by reducing the matrix from N space to K space where K < N. For the point of the guidance framework be that as it may, we are just interested about by the matrix factorization part keeping same dimensionality. The matrix factorization is completed on the user-item ratings matrix built.

Each item is often represented with a q vector. Similarly, each client are often represented by a p vector such the inner product of these 2 vectors is that the expected rating.

Find p and q such it minimizes the following:

$$expected\ rating = \widehat{r_{ui}} = q_i^T p_u$$

$$minimum\ (p,q) \sum_{(u,i)\in K} (r_{ui} - q_i^T . p_u)^2$$

$$minimum\ (p,q) \sum_{(u,i)\in K} (r_{ui} - q_i^T . p_u) + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

For our model to be able to generalize well and not over fitting the training set, we introduce a penalty term our minimization equation. This is often represented by a regularization factor multiplied by the sum of the Squares of magnitudes of user and item vectors.

B. Nearest Neighborhood Model

Nearest Neighborhood model involves collecting data from numerous clients at that point making forecasts according to the similarity measures between users and between items. This might be grouped into client-based and item based models. In item-based model, it's expected that songs that are frequently listened together by certain clients will in general be indistinguishable and are bound to be listened together in future additionally by another client. According to client based similarity model, clients who have similar listening histories, i.e., have listened in to similar songs inside the past will in general have comparative interests and may most likely hear similar songs in future as well.

We need some similarity measure to coordinate between two songs or between two clients. Cosine similarity gauges every one of the clients similarly which is generally not the situation. Client ought to be weighed less if he has shown interests an excessive amount of kind of items (it shows that it is possible that she doesn't perceive between songs supported their quality, or just prefers to explore). Similarly, client is weighted more if tunes in to restricted set of songs. The similarity measure,

wij = P(i /j),

additionally has drawbacks that a few songs which are listened more by clients have higher similarity esteems not on the grounds that they're similar and listened together but since they're increasingly well known.

IV. ABOUT DATASET AND ATTRIBUTES

The examination was supervised on Million songs data set, which is additionally posted on Columbia University for assessing the precision and presentation of machine learning techniques. The dataset consists of around 48 million triplets/features collected from histories of over a million users and metadata of many songs i.e.280 GB and it contains attributes information about various songs and attributes like userid, songid, play count, etc. the info set is split into trained data and test data: 80% of coaching data and 20% of testing data.

Count number of unique users in the dataset

```
users = song_df['user_id'].unique()
len(users)
```

```
365
```

Count the number of unique songs in the dataset

```
songs = song_df['song'].unique()
len(songs)
```

```
5151
```

Fig 2:- Count Number of unique users and songs in the dataset.

V. RESULTS AND ANALYSIS

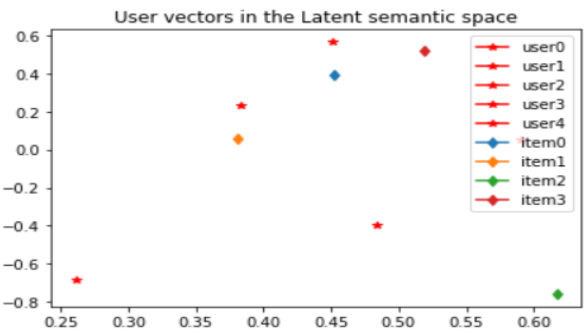

Fig 3:- Understanding in tuition behind SVD.

ISSN No:-2456-2165

```python
#Compute estimated rating for the test user
def computeEstimatedRatings(urm, U, S, Vt, uTest, K, test):
    rightTerm = S*Vt

    estimatedRatings = np.zeros(shape=(MAX_UID, MAX_PID), dtype=np.float16)
    for userTest in uTest:
        prod = U[userTest, :]*rightTerm
        #we convert the vector to dense format in order to get the indices
        #of the movies with the best estimated ratings
        estimatedRatings[userTest, :] = prod.todense()
        recom = (-estimatedRatings[userTest, :]).argsort()[:250]
    return recom
```

```python
#Used in SVD calculation (number of latent factors)
K=2

#Initialize a sample user rating matrix
urm = csc_matrix(user_song_df.values, dtype=np.float32)

#Compute SVD of the input user ratings matrix
U, S, Vt = computeSVD(urm, K)

uTest = [342]
print("User id for whom recommendations are needed: %s" % uTest[0])
#print(U)
#print(Vt)
#Get estimated rating for test user
print("Predictied ratings:")
uTest_recommended_items = computeEstimatedRatings(urm, U, S, Vt, uTest, K, True)
print(uTest_recommended_items)
```

Fig 4:- Input code for User Prediction using SVD.

```
User id for whom recommendations are needed: 342
Predictied ratings:
[ 331 1588  763 3642 2844 3766 3599  704 4687  435 2817 2566 3235 4254
 2391 1739 3719 4175 2120 3436 2722 3812 4365 2437  708 4468  166 3346
 3056 2970 1860 3016 2850  553 1152 3240  518 1738  616  436 3992  186
  655 2546 2662 4316 3782 3210 2265 3545  485 1718 2006 2607 3784 4136
 3517   29 2331 3222  897 1923 1768 4617 1040 4011  710 3555 4234  613
 3952 3213 1158 2501 2691   55 4122 2712  387  325 3951 1778 4694 4615
 3391  162  668 3438 3875 1401 5036 2416 2290 1476 4014 4933 4599 2765
 4774 4453  152  720 4640  132 3280  427 1902 5042  412 1322  253 3452
 3838 1137  867 4337  958  640 2991 4749 4072 2935 4407 1798 4609 2931
 2498 4251 1959  667 1729 5089 1229  415 4376 1140 4496 2206 2470  189
 3447 1681 4917 4226 1068 2710 4399 2671 1217 4073 4141 4108 1465 4149
 2389 4696 1648 3276 1091  741  681 3572 1373 5146 1174  300 3649 1973
 1992   72 1135 4999 4386  105 3628 1351  329 2893  818 3957 1870 1222
 2337  657 4012 3242 2836 5101 2007 1016 4056   40  209 2199 3884 2378
 3699 1039 1585 4357  749 1531 2070 2842 4206 5004 1374  475 1114   92
  492 2052 3790 2192 3729 3117 2287 2280 2952 4004 2507 1938  874 4515
  470 2394 1070 1632 2752 4466  682 4172 4964 3520 2592  601  151  290
 2142 4726 2499 4579  915 1935  532  954 3127 2370 3716 3414]
```

Fig 5:- Output Prediction for Single User.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we explain a basic metadata-based model and two famous music genre recommender approaches: collaborative altering and content-based model. In spite of the fact that they need made incredible progress, their drawbacks like popularity bias and human-efforts are self-evident. Moreover, the utilization of hybrid model would outperform one model since it incorporates the benefits of both methods. Its complexity isn't fully studied yet, thanks to the subjective nature in music and therefore the issues existing within the previous methods, two human-centered approaches are proposed. By thinking about affective and social data, emotion based model and context based model generally improved the standard of exhortation. However, this research remains at a beginning stage.

REFERENCES

[1]. Soleymani, M., Aljanaki, A., Wiering, F., & Veltkamp, R. C. (2015, June). Content-based music recommendation using underlying music preference structure. In Multimedia and Expo (ICME), 2015 IEEE International Conference on (pp. 1-6). IEEE.

[2]. Makarand, V., & Sahasrabuddhe, H. V. (2014, January). Novel Approach for Music Search Using Music Contents and Human Perception. In Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on (pp. 1-6). IEEE.

[3]. Everyone listens to music, but how we listen is changing. Available at: http://www.nielsen.com/us/en/insights/news/2015/everyone-listens-to-music-but-how- we-listen-is-changing.html

[4]. Labrosa.ee.columbia.edu. (2017). Million Song Dataset | scaling MIR research. [online] Available at: https://labrosa.ee.columbia.edu/millionson/

[5]. Lee, J., Shin, S., Jang, D., Jang, S. J., & Yoon, K. (2015, January). Music recommendation system based on usage history and automatic genre classification. In Consumer Electronics (ICCE), 2015 IEEE International Conference on (pp. 134-135). IEEE.

[6]. Reddy, M. S., & Adilakshmi, T. (2014, January). Music recommendation system based on matrix factorization techniqueSVD. In Computer Communication and Informatics (ICCCI), 2014 International Conference on (pp. 1-6). IEEE.

[7]. Object detection. Available at: https://towardsdatascience.com/review-yolov2-yolo9000-you-only-look-once-object-detection-7883d2b02a65