

MUSIC RECOMMENDATION SYSTEM USING PYTHON



**A Course Project report submitted in partial fulfillment of the Degree of
BACHELOR OF TECHNOLOGY**

in

COMPUTER SCIENCE AND ENGINEERING

By

18K41A0599

Sachin kumar Koppula

18K41A0596

Deekshitha Muppidi

18K41A0536

Anitha More

18K41A04C3

Sindhu Chinthinreddy

Under the Guidance of

Mr . D.Ramesh

(Assistant Professor)

Submitted to



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SR ENGINEERING COLLEGE**

Ananthasagar, Hasanparthy (M), Warangal Urban, Telangana 506371

(Affiliated to JNTUH, Accredited by NBA) July, 2019.

SR ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Course Project Report entitled “**Music Recommendation System**” is a record of bonafide work carried out by the student(s) **Sachinkumar Koppula , Deekshitha Muppidi , Anitha More, Sindhu Chinthinreddy** bearing Roll No(s) **18K41A0599, 18K41A0596, 18K41A0536, 18K41A04C3** during the academic year 2019-20 in partial fulfillment of the award of the degree of *Bachelor of Technology* in **Computer Science & Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

Supervisor

Head of the Department

External Examiner

Abstract :

In this project, we planned, executed and evaluated the song Recommendation mechanisms using a number of algorithms. Music recommendation is a very complicated issue when we need to arrange music in such a manner that we recommend the favorite songs to consumers that is never a definitive forecast. It is diverse and often affected by influences other than the listening history of users or songs. We used dataset available in static turi and kaggle, to find correlations between users and songs, and to benefit from the user's past listening experience to provide recommended songs that users will likely listen to. We're going to Discuss the challenges we have encountered, the processes we have put in motion, the outcomes and the analysis. We also obtained the best results using a shared neural network filtering algorithm.

Table of contents :

S.No	CONTENTS	Page No(s)
1.	Abstract	3
2.	Table of Contents	4
3.	Introduction	5
4.	Literature Review	7
5.	Design	8
6.	Implementation	9
7.	Results	13
8.	Accuracy	19
9.	Conclusion	19
10.	References	20

Introduction :

The Music Recommendation System is a system that learns about the user's previous listening experience and recommends them with tracks that they'd probably want to hear in the future. We also put in place different algorithms to try to create an accurate framework of recommendations. First, we applied a popularity-based paradigm that was simple and intuitive. Collaborative filtering algorithms that predict the user's taste (filtering) by gathering expectations and tastes from several other users (collaborating) are now introduced. We've also done neuronal collective filtering studies to determine the best the parameters of the model.

Our simulations will attempt to analyze and minimize the error between the expected ratings and the real ratings. Recommendation mechanisms are perhaps the most normally actualized AI calculations. The suggested (or recommended) method (or engine) is a filtering system that is programmed to anticipate the rating or choice that the user will provide to an object, e.g. A video, a product, a song, etc.

There are two major types of recommended systems:

- Content-based filters
- Collaborative filters

For attaining this goal we use specific steps:

1. Problem Statement
2. Data Set
3. Pre Processing , dividing data into train and test datasets and Visualising
4. Applying Recommendation Algorithms
5. Conclusion

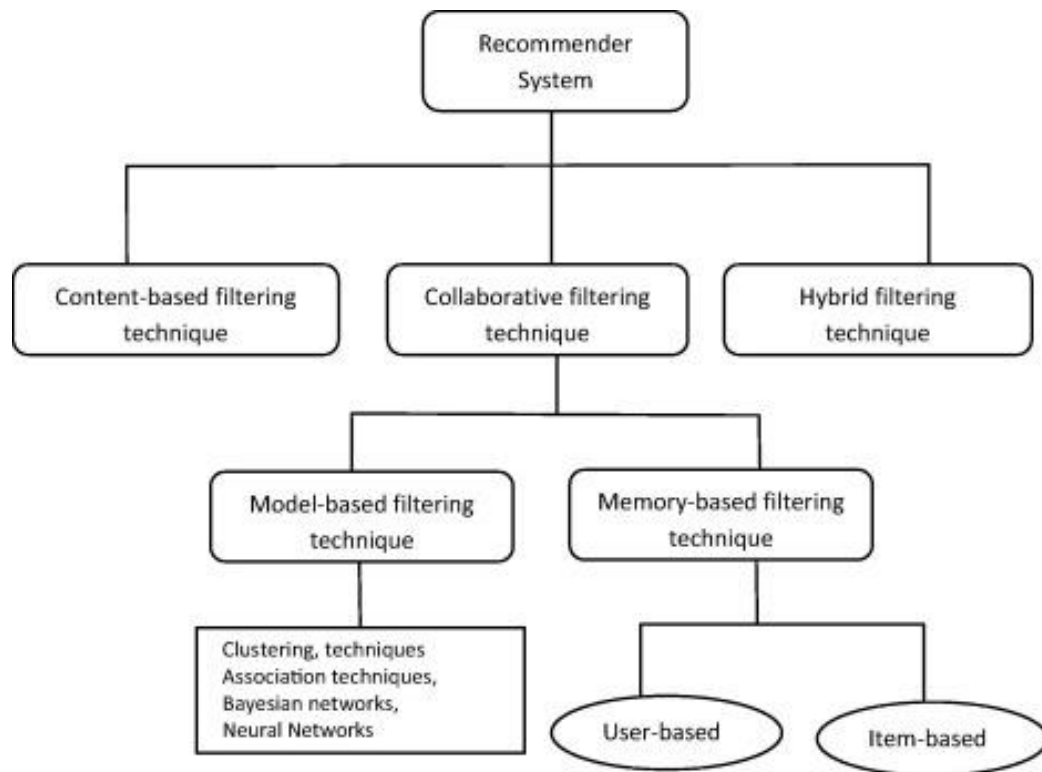


Figure : Classification of the different recommendations algorithms used in this project

Literature Review :

Current recommended systems using collaborative filtering algorithms have achieved considerable success. Netflix announced prize amount to the best collaborative filtering algorithm, and the winning algorithm using latent factor models will allow 10.09 percent progress over the algorithm used by Netflix at the time. Amazon employs collective filtering focused on user-user and item-item, which significantly contributes to the performance of the enterprise. Newer algorithms using neural network, collective neural filtering, have recently been suggested.

Many researchers have suggested various approaches for content-based algorithms using Machine Learning techniques, such as Decision Tree-based, Vector-based support, and even logistic regression. We will make good use of the skills we have gained from the class to apply these algorithms.

The Music Recommendation System has some parallels with other commercial recommendation services, but relies more on offering good and customized music advice rather than products for consumers to order. The optimal music-recommending device should be able to instantly recommend customized music to human listeners. Unlike books or film, the length of a piece of music is much shorter, and the occasions when listening to their favorite songs are typically more than once, which are the key difficulties that we will encounter in this project.

We will recommend music based on popularity, similarity and use both content based and collaborative based and collaborative based that we have used Matrix factorization approach with SVD, k-nearest algorithm approach. and recommend music by decreasing RMSE.

Design :

This is a recommendation system that suggests a user-based past history of songs listened to by users and songs that are trending, i.e. listened to by most users using various machine learning tools such as pandas, matplotlib, sckit-learn, etc.

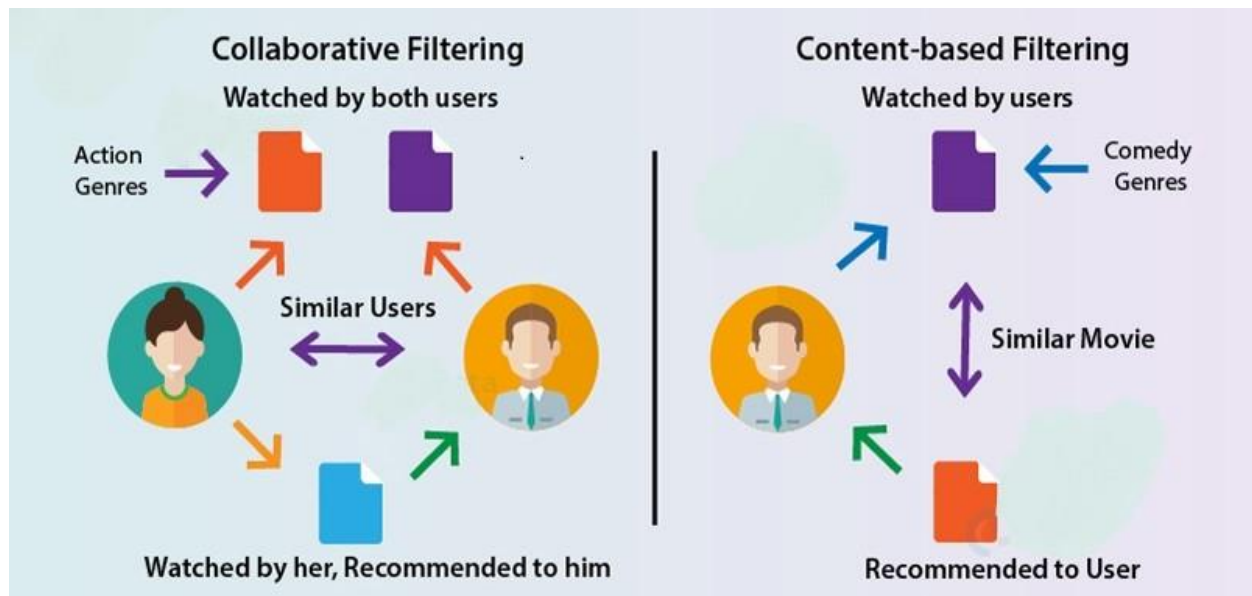


Figure : Procedure of recommendations systems generally in both the ways

Implementation :

Dataset :

We extracted data from the Million Song Dataset from static turi and kaggle. which contains two files songs data, user data and songs dataset for content based filtering from kaggle

1. Songs data (1000000 rows) attributes :
 - Song_id
 - Title
 - Release
 - Artist_name
 - Year
2. User data attributes :
 - User id
 - Song id
 - Listen count
3. Songs data attributes for content based filtering :
 - Artist
 - Song
 - Link
 - Text

Algorithms :

We use different machine learning model to solve our classification problem:

1. Content-based filters
2. Collaborative filters
 - K-Nearest Neighbors (kNN)
 - matrix factorization approach.

So, Let's make our data ready for training and testing our machine learning model.

Content-based filters :

Recommendations developed using this filters method are user-specific classification problem. This classifier knows what the user wants and does not like the features of the song.

The most simple solution is matching keywords. The idea behind this is to extract important keywords which are present in a song description that the user likes very much, to look for keywords in other song descriptions to approximate the similarities between them and, on that basis, to recommend those songs to the user.

How is this done? In our case, since we're dealing with text and terms, Term Frequency-Inverse Document Frequency (TF-IDF) can be used for this matching process.

We're going through the steps to build a content-based music-recommended structures.

K-Nearest Neighbour (KNN) :

Collaborative Filters work with an interaction matrix, also called rating matrix. The purpose of this algorithm is to learn a feature that can predict when a user would benefit from an object-meaning that the user is likely to order, listen to, and watch this item.

There are 2 types of collaborative-based recommendation systems:

- a. user-item filtering and
- b. item-item filtering.

There are several machine learning algorithms that can be used for collective filtering. Among them, we can list the nearest-neighbor, clustering, and matrix factoring.

K-Nearest Neighbors (kNN) is considered a standard framework for collaborative filtering approaches based on both the recipient and the object.

There are two files that will be interesting for us. The first of them will give us information about the songs. Particularly, it contains the user ID, song ID and the listen count. On the other hand, the second file will contain song ID, title of that song, release, artist name and year. We need to merge these two DataFrames. For that aim, we'll use the song_ID

Popular songs :

we'll count how many times each song appears. Note that while we are using `listen_count`, we only care about the number of rows, we don't consider the number present in that row. This number represents how many times one user listen to the same song

Matrix factorization method (using SVD) :

Collaborative Filters operate with an interaction matrix, also referred to as a ranking matrix. The aim of this algorithm is to learn a function that can predict when a user will profit from an object—meaning that the user is likely to request, listen to, and watch this thing.

There are two types :

- a. user-item filtering and
- b. item-item filtering.

We're going through the steps to create a music recommendation framework. We're going to use a matrix factorization method this time.

Once again, we will use the Million Song Dataset, a freely distributed catalog of audio attributes and metadata for a million contemporary popular music songs that we used for the KNN methodology

Matrix Factorization is a strong means of applying a recommendation framework. The concept behind this is to represent users and objects in a latent space , lower-dimensional Kspace.

In other words, matrix factorization methods decompose the original sparse user-item matrix into smaller, less sparse rectangular matrixes with latent functionality.

This not only solves the problem of sparsity, but also makes the process scalable. No matter how large the matrix is, you will still find a lower dimensional matrix that is a true reflection of the original matrix.

Among the many matrix factorization techniques, we considered the popular SVD where `svd` : singular value decomposition

This may be an abstract term as we expand our comprehension of the mathematical foundations. But we're going to try to make things as easy as possible. Imagine that we have a matrix A that contains data for users of n albums.. This matrix can be decomposed into three matrixes only; let's name them U , S , and V .

As far as song recommendation is concerned:

U is a n user \times r user-latent function matrix

V is a m song \times r song-latent function matrix

S is a non-negative diagonal matrix of $r \times r$ containing the singular values of the original matrix.

Instead of dealing with the tacit ranking as it is, we can use the binning method.

We will classify all of them into 10 classes. The original data values that fall within the range of 0 to 1 will be replaced by the symbolic ranking of 1; if they fall within the range of 1 to 2, they will be replaced by 2; and so on and so on. The last category will be assigned to the initial values from 9 to 2213

We'll use the built-in function for SVD. First, a set of parameters is going to be defined to search for the best parameters for the model.

The GridSearchCV class will compute accuracy metrics for the SVDalgorithm on the combinations of parameters selected, over a cross-validation procedure. This is useful for finding the best set of parameters for a prediction algorithm.

Results :

Data Processing :

	user_id	song_id	listen_count	title	release	artist_name	year
0	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOAKIMP12A8C130995	1	The Cove	Thicker Than Water	Jack Johnson	0
1	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBBMDR12A8C13253B	2	Entre Dos Aguas	Flamenco Para Niños	Paco De Lucia	1976
2	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBXHDL12A81C204C0	1	Stronger	Graduation	Kanye West	2007
3	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBYHAJ12A6701BF1D	1	Constellations	In Between Dreams	Jack Johnson	2005
4	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SODACBL12A8C13C273	1	Learn To Fly	There Is Nothing Left To Lose	Foo Fighters	1999

Figure : After merging userdata and songs data

Cleaning of data :

we should perform some cleaning steps. But looking at the dataset, we can see that there is no missing values. And most of the columns contain strings

```
1 df_songs.isnull().sum()

user_id      0
song_id      0
listen_count  0
title        0
release      0
artist_name  0
year         0
dtype: int64
```

```
1 df_songs.dtypes

user_id      object
song_id      object
listen_count  int64
title        object
release      object
artist_name  object
year         int64
dtype: object
```

Figure : Cleaning of dataset

Content-based filters :

We pick random song from dataset and test it

```
1 recommendation = {  
2     "song": songs['song'].iloc[10],  
3     "number_songs": 4  
4 }
```

```
1 recommendations.recommend(recommendation)
```

The 4 recommended songs for Buck Dich (English) are:

Number 1:

25 by Veruca Salt with 0.403 similarity score

Number 2:

Lovely To See You by Moody Blues with 0.293 similarity score

Number 3:

Is It Me Is It You by Lenny Kravitz with 0.187 similarity score

Number 4:

Already There by Verve with 0.162 similarity score

Another random songs picked

```
1 recommendation2 = {  
2     "song": songs['song'].iloc[120],  
3     "number_songs": 4  
4 }
```

```
1 recommendations.recommend(recommendation2)
```

The 4 recommended songs for Man Smart, Woman Smarter are:

Number 1:

Hold What You've Got by Hank Williams Jr. with 0.22 similarity score

Number 2:

Wild Women Do by Natalie Cole with 0.216 similarity score

Number 3:

Honky Tonk Songs by Dolly Parton with 0.214 similarity score

Number 4:

Saw Mill (set I) by Grateful Dead with 0.143 similarity score

K-Nearest Neighbour (KNN) :

	title	listen_count	percentage
6836	Sehr kosmisch	8277	0.41
8725	Undo	7032	0.35
1964	Dog Days Are Over (Radio Edit)	6949	0.35
9496	You're The One	6729	0.34
6498	Revelry	6145	0.31
6825	Secrets	5841	0.29
3437	Horn Concerto No. 4 in E flat K495: II. Romanc...	5385	0.27
2595	Fireflies	4795	0.24
3322	Hey_ Soul Sister	4758	0.24
8494	Tive Sim	4548	0.23

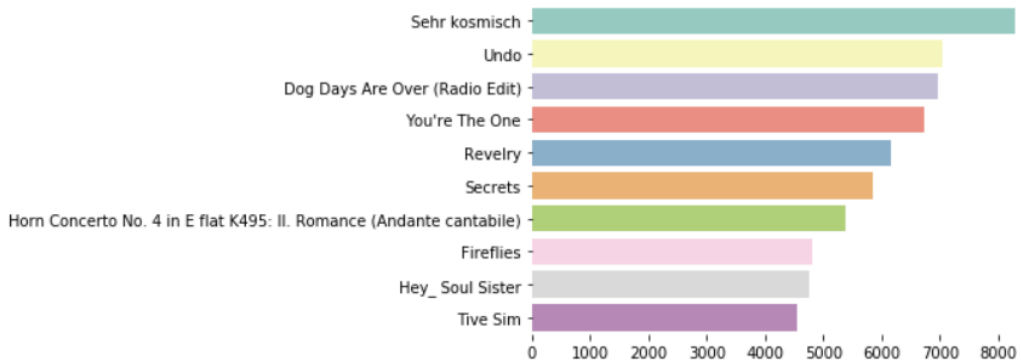


Figure : Graphical result of popular songs

Popular artists :

	artist_name	listen_count
649	Coldplay	29422
2850	The Black Keys	19862
1651	Kings Of Leon	18747
1107	Florence + The Machine	18112
1370	Jack Johnson	17801
2946	The Killers	16063
2374	Radiohead	14890
736	Daft Punk	14715
2073	Muse	14005
1554	Justin Bieber	13959

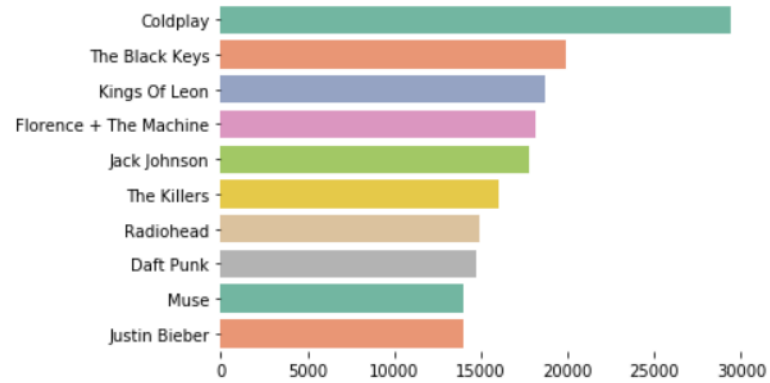


Figure : Graphical result of popular artist

We can also get some other information from the feature listen_count :

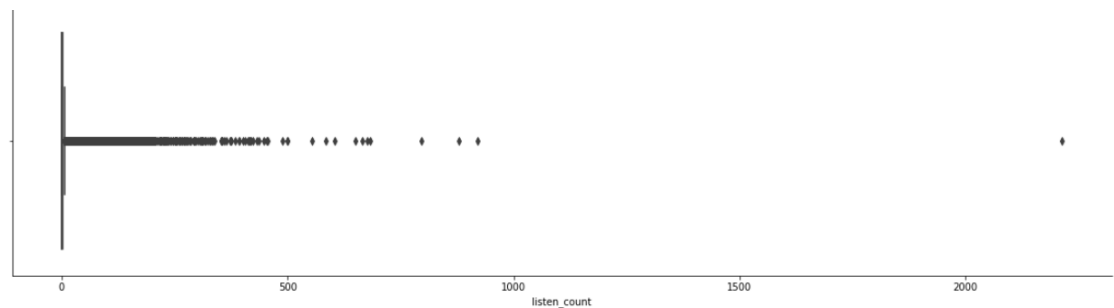


Figure : distribution of listen_count

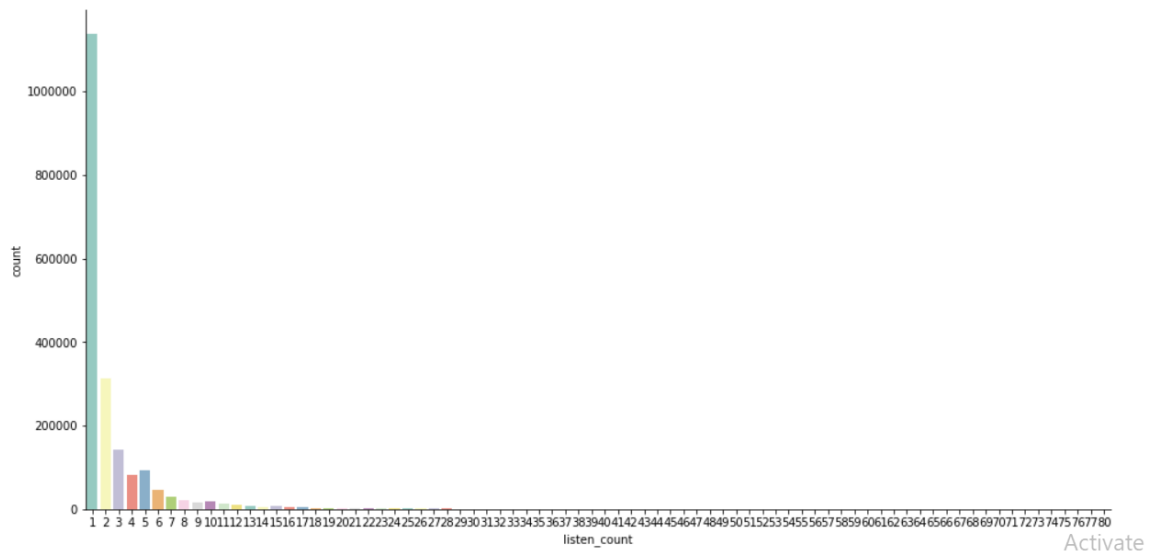


Figure : most frequent number of times a user listen to the same song

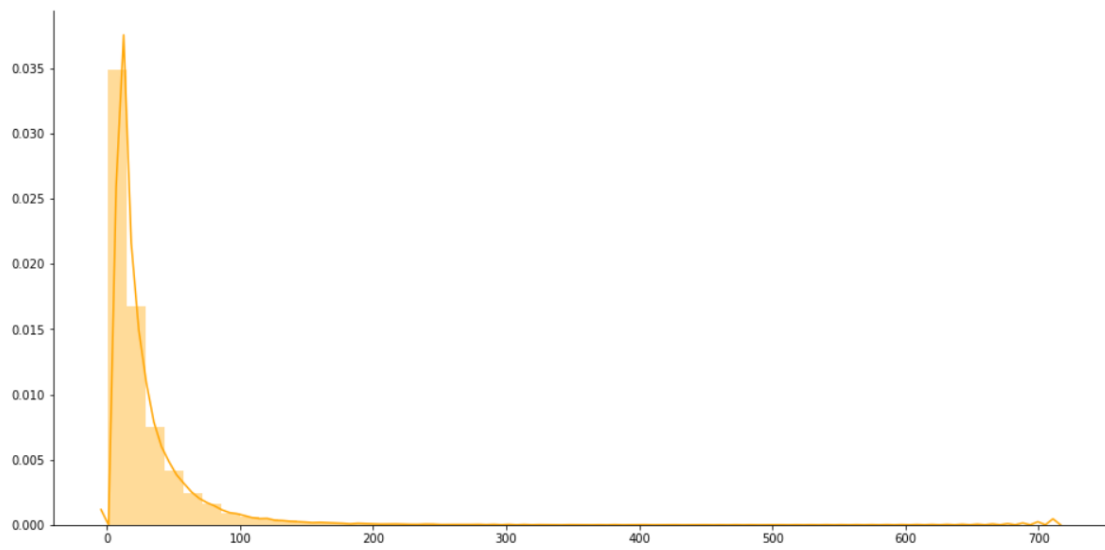


Figure : songs user listen in average

Matrix factorization method (using SVD) :

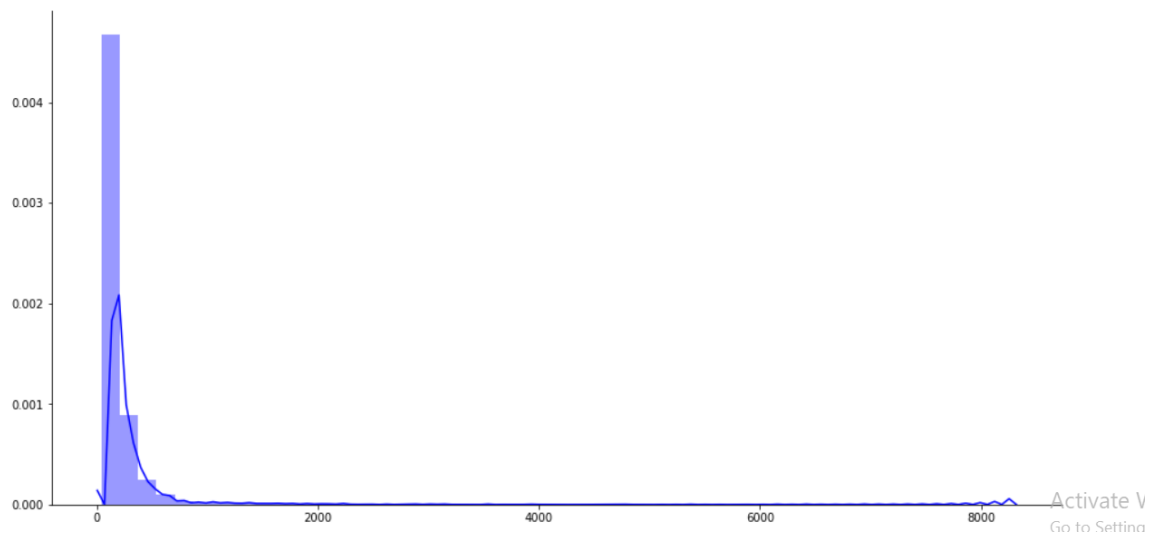


Figure : number of users listen to same song on average

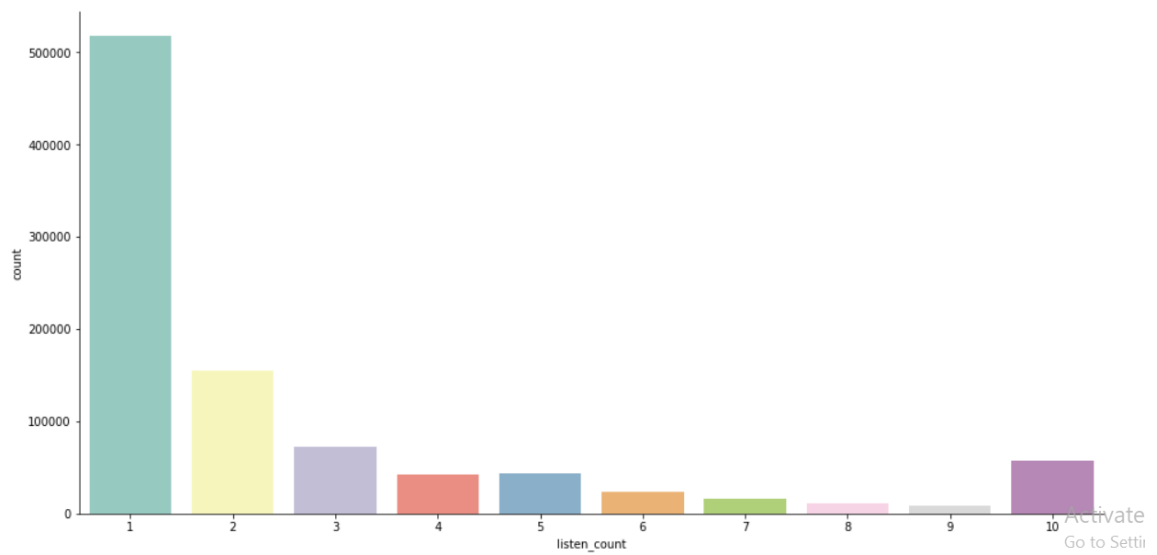


Figure : Listen count of categories divided

Accuracy :

	Fold1	Fold2	Fold3	Fold4	Fold5	Mean	Std
RMSE (testset)	2.1709	2.1768	2.1832	2.1807	2.1653	2.1754	0.0065
Fit(train) time	860.41	852.93	800.15	462.58	459.12	687.04	185.85
Test time	6.27	5.17	3.03	2.36	2.73	3.91	1.53

Table: Testing the RootMeanSquareError of the SVD algorithm with 5 splits.



RMSE: 2.1781
The RMSE is 2.1780687698192116

Figure : RMSE of SVD algorithm

Conclusion :

So, From the above dataframe we can predict that matrix factorisation using SVD Model is result into best score among the three different model. But we try to make our others model accuracy more accurate but at this stage we can see that our problem scoring only with 2.178 RMSE which is given by matrix factorisation using SVD algorithm.

References:

1. <https://static.turi.com/>
2. They're all listening to music, but how we listen is changing. [online] Available at: <http://www.nielsen.com/us/insights/news/2015/everyone-listen-to-music-but-how-listen-is-changing.html> [Accessed 10 Oct. 2017].
3. Labrosia.ee.columbia.edu. Million Song Dataset | Scale MIR Study Available at: <https://labrosia.ee.columbia.edu/millionsong/> (Accessed 10 Oct [2017]).
4. En. wikipedia.org. [2017]. The Netflix Award. Available at: https://en.wikipedia.org/wiki/Netflix_Prize [Accessed 11 October 2017].
5. Linden, G., Smith and B. And York, J. (2003). Amazon.com Recommendations Collaborative Filtering Item-to-Item. Available at the following: <https://www.cs.umd.edu/~samir/498/Amazon-recommendations.pdf>.
6. Xia Hu, Liqiang Nie, Tat Seng Chua, and Lizi Liao, Hanwang Zhang, Lizi Liao . 2017. Neural Collaborative Filtration. In the proceedings of the 26th World Wide Web International Conference (WWW'17). International World Wide Web Conference Steering Committee, Republic and Canton of Geneva, Switzerland, 173-182. DOI: <https://doi.org/10.1145/3038912.3052569>
7. Gershman, Gershman, A. And Meisels... And Meisels, A. (2015). (2015). A Tree-based Suggestion Framework for Judgment. Pp.170-179, IEEE Xplore. Offered at: The following copies are available: https://subs.emis.de/LNI/Proceedings/Proceedings_ings165/170.pdf [Accessible 9 Oct. 2017].
8. Brian McFee, Gert R. G. Lanckriet. Learning to rate metrics. At the 27th International Machine Learning Conference, 2010.
9. R-Bloggers, [2017]. Hybrid content based and interactive filtering Recommendation with {ordinal} logistic regression (2): recommendation as a confidential alternative. [online] Available at: <https://www.r-bloggers.com/hybrid-based-content-and-collaborative-filtering-recommendations-with-ordinal-logistic-regres>
10. [8] Han I, Min SH. (2005) Using Support Vector Machines, Recommender Networks. In: D., Gaedke M. In: (eds) Software computing. Web engineering. 2005 ICWE. Data Science Reading Instructions, vol 3579. Berlin, Springer, Heidelberg Hills

11. Viet-Anh Tran et al. "Making Effective Negative Sampling Collaborative Metric Learning." At: the 42nd International ACM SIGIR Conference on Knowledge Recovery Research and Development. PRODUCTION 19. ACM 2019, pp. 1201–1204, Paris. isbn: 978 1 4503 6172 9. doi : 10.1145/3331184.3331337. url : <http://doi.acm.org/10.1145/332184.331338>.
12. Kazuyoshi Yoshii, Masataka Goto and Hi-roshi G Okuno, Kazunori Komatani. Incremental preparation focused on improving the performance and scalability of model music referees. The 8th International Music Information Conference Retrieval, ISMIR Amount, 2007, held in ISMIR.
13. Vignoli, Fabio. A user-driven similarity-based music recovery framework and its evaluation. Retrieval of Music Knowledge 2005 at the International Conference.2005.
14. And Perfecto Herrera and Dmitry Bogdanov. How much metadata do we need in music? The use of Choice Sets for subjective evaluation. ISMIR2011, pages 97–102, 2011 at the 12th International Society for the Recovery of Music Content Meeting.
15. G. A. and Adomavicius. Tushilina. Tushiline. RecommendingSystems: A survey of state-of-the-art and future expansions into the next decade. Information and Data Engineering transactions of the IEEE, 17(6):734–749, June 2005