**Graphic Era Deemed to be University**

**Dehradun, Uttarakhand**

**Logo

Description automatically generated**

**A Mini Project Report on**

**Music Recommender System**

**(*Machine Learning*)**

Name: Nilesh Popli

University Roll No.: 2016881

Course: B. Tech (CSE)

Semester: 4th

Guided By:

**ACKNOWLEDGEMENT**

Hereby I am submitting the project report on MUSIC RECOMMENDER SYSTEM, as per the scheme of Graphic Era University, Dehradun.

In this connection, I would like to express my deep sense of gratitude to our beloved institution Graphic Era University and, I like to express our sincere gratitude and indebtedness to Prof. (Dr). Kamal Ghanshala, founder of GEU, Dehradun.

I take this opportunity to express our sincere gratitude to Ms. Vishu Tyagi Mam for her valuable guidance in this project. Without her, it would not have been completed. The report has been developed as a result of his valuable advice.

Also, I am thankful to my classmates and friends for their support and guidance. I also like to thank researchers and scholars whose papers and thesis have been utilized in this report.

**TABLE OF CONTENTS**

* **Introduction**
  + Abstract
  + Motivation
* **Literature Review**
  + Recommendation System
  + Dataset Used
* **Methodology**
  + Popularity Based Model
  + Collaborative Filtering
  + Evaluation Metrics
* **Conclusion**
  + Summary
  + Future Scope
* **Bibliography/ Reference**

**INTRODUCTION**

**Abstract**

With commercial music streaming service which can be accessed from mobile devices, the availability of digital music currently is abundant compared to previous era. Sorting out all this digital music is a very time-consuming and causes information fatigue. Our research would like to develop a music recommender system that can give recommendations based on similarity of features on audio signal. This study uses convolutional recurrent neural network (CRNN) for feature extraction and similarity distance to look similarity between features.

**Motivation**

With the rise of digital content distribution, people now have access to music collections on an unprecedented scale. Commercial music libraries easily exceed 15 million songs, which vastly exceeds the listening capability of any single person. With millions of songs to choose from, people sometimes feel overwhelmed. Thus, an efficient music recommender system is necessary in the interest of both music service providers and customers. Users will have no more pain to make decisions on what to listen while music companies can maintain their user group and attract new users by improving users’ satisfaction. In the academic field, the domain of user centric music recommendation has always been ignored due to the lack of publicly available, open and transparent data. Million Song Dataset Challenge provides data which is open and largescale which facilitates academic research in user centric music recommender system which hasn’t been studied a lot.

**LITERATURE REVIEW**

**What is a Recommendation System?**

A Recommendation System aims to predict the user’s choices and recommend the product or service that is likely to be interesting.

These systems can do so because of user data. The function of a Recommendation System mainly depends on two kinds of information:

1. Characteristic information: Information that defines the profile of a product (tag, category, etc.) or a user (preferences, profile, etc.).
2. User-item interactions: Information that defines user-item relationship (rating, like/dislike, etc.).

![Diagram, engineering drawing

Description automatically generated]()

**Dataset used**

I have used data provided by Million Song Data Challenge hosted by Kaggle. The Million Song Dataset Challenge aims at being the best possible offline evaluation of a music recommendation system. Any type of algorithm can be used: collaborative filtering, content-based methods, web crawling, even human oracles!\* By relying on the Million Song Dataset, the data for the competition is completely open: almost everything is known and possibly available.

The Million Song Dataset Challenge is a joint effort between the Computer Audition Lab at UC San Diego and LabROSE at Columbia University. The user data for the challenge, like much of the data in the Million Song Dataset, was generously donated by The Echo West, with additional data contributed by Secondhandsongs, musiXmatch, and Last.fm.

![Table

Description automatically generated]()

![A picture containing table

Description automatically generated]()

**METHODOLOGY**

**Popularity Based Model**

It is the most basic and simple algorithm. I find the popularity of each song by looking into the training set and calculating the number of users who had listened to this song. Songs are then sorted in the descending order of their popularity. For each user, we recommend topmost popular songs. This method involves no personalization, and some songs may never be listened in future.

**Collaborative Filtering**

Graphical user interface

Description automatically generatedCollaborative filtering systems use user-item interactions to generate recommendations. This means collaborative filtering uses similarities between users and items simultaneously to provide recommendations. Let us refer to the diagram below.

As we can see in the diagram, there are two users, user A, and user B. Both users have similar tastes in music as both liked song-1 and song-2, but there is a song-3 which user A likes, but user B never listened to it. The system will recommend song-3 to user A based on these user-item interactions.

We need some similarity measure to compare between two songs or between two users.

We have used conditional probability-based model of similarity between users and between items:

Wu,v = P(v/u) αP(v/u) 1−α, α(0, 1)

Wuv = |I(u) T I(v)| |I(u)| α|I(v)| 1−α [2]

Then, for each new user u and song i, user-based scoring function is calculated as

h U ui = P vui f(wuv)[2]

Similarly, item-based scoring function is

h I ui = P jIi f(wij )[2]

**Evaluation Metrics**

We used mean Average Precision (mAP) as our evaluation metric. The reason behind using this is that this metric was used in the Kaggle challenge which helps us to compare our results with others. Moreover, precision is much more important than recall because false positives can lead to a poor user experience. Our metric gives the average of the proportion of correct recommendations giving more weight to the top recommendations. There are three steps of computing mAP as follows:

Firstly, precision at each k is calculated. It gives the proportion of correct recommendations within the top-k of the predicted rankings.

Pk(u, r) = 1 k Pk j=1 M(u, r(j))

Then, for each user, average precision at each k is evaluated.

AP(u, r) = 1 nu Pt k=1 Pk(u, r).M(u, r(k))

Finally, mean of all the users is taken

mAP = 1 m Pm u=1 AP(u, ru)

**CONCLUSION**

**Summary**

From this Project I learned very much before this project I was very scared that how I complete it but slowly slowly everything become easy for me. When I faced any difficulty, I use google for solving my problems. I learned new things from this project which boosts my skills and knowledge. All these difficulties due to the data and to the system itself make it more challenging and more attractive. I hope that I will get other opportunities in the future to work in the domain of artificial intelligence. I am certain that I can do a better job.

**Future Scope**

This project can be improved in the future, by keep updating the dataset by adding songs. Also, it will be better to use live dataset for the same.

We can link this project with any Music Application like Spotify, Resso and more, then using their playlist, dataset, songs, we can not only recommend but can easily play songs from them. We can collaborate this project to work with them.

Also, a GUI could be made, to make this more attractive and easier to use.

**REFERENCES/ BIBLIOGRAPHY**

1. Dataset: <https://www.kaggle.com/c/msdchallenge/data>
2. <https://www.geeksforgeeks.org/project-idea-songs-recommendation-system-in-android/>
3. <https://www.geeksforgeeks.org/machine-learning/>
4. https://www.udemy.com/course/python-programming-beginner-to-advanced/
5. <https://pythonprogramming.net>