International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT217615

Music Recommendation System Using Machine Learning

Varsha Verma, Ninad Marathe, Parth Sanghavi, Dr. Prashant Nitnaware
Department of Information Technology, PCE, Navi Mumbai, Maharashtra, India

Article Info
Volume 7, Issue 6
Page Number: 80-88
Publication Issue :
November-December-2021
Article History
Accepted : 10 Nov 2021
Published : 20 Nov 2021

ABSTRACT

In our project, we will be using a sample data set of songs to find correlations between users and songs so that a new song will be recommended to them based on their previous history. We will implement this project using libraries like NumPy, Pandas.We will also be using Cosine similarity along with CountVectorizer. Along with this,a front end with flask that will show us the recommended songs when a specific song is processed.

Keywords : Numpy, Pandas, Cosine Similarity, Count Vectorizer

I. INTRODUCTION

With the explosion of networks in the past decades, the internet has become the major source of retrieving multimedia information such as video, books, and music, etc. People have considered that music is an important aspect of their lives and they listen to music, an activity they engage infrequently. People sometimes feel it is difficult to choose from millions of songs. With commercial music streaming services which can be accessed from mobile devices, the availability of digital music currently is abundant compared to the previous era. Music service providers need an efficient way to manage songs and help their customers to discover music by giving quality recommendations.

A music recommender system is a system that learns from the user's past listening history and recommends songs which they would probably like to hear in the future. By using a music recommender system, the music provider can predict and then offer the appropriate songs to their users based on the characteristics of the music that has been heard previously. Sorting out all this digital music is very time-consuming and causes information fatigue. Therefore, it is very useful to develop a music recommender system that can search in the music libraries automatically and suggest suitable songs to users. Thus, there is a strong need for a good recommendation system.

Recommendation Systems are everywhere and pretty standard all over the web. Currently, there are many music streaming services, like Pandora, Spotify, etc., which are working on building high-precision commercial music recommendation systems. Amazon, Netflix, and many such companies are using Recommendation Systems. Music recommendation is a very difficult problem as we have to structure music in a way that we recommend the favorite songs to users which is never a definite prediction. In this project, we have designed, implemented, and analyzed a song recommendation system. The one we are going to build is pretty common to what Spotify or Youtube Music uses but much more straightforward. Currently, most of the streaming music systems recommend songs based on Collaborative Filtering and Content-Based filtering techniques

While collaborative filtering (CF) has been the most common choice in those early days of RS research, approaches based on content-based filtering (CBF) have gained popularity in recent years. In short, collaborative filtering approaches exploit interactions between users and items, e.g., clicks or ratings, which are represented in a user-item (rating) matrix R.

Collaborative filtering System: Collaborative does not need the features of the items to be given. Every user and item is described by a feature vector or embedding. It creates embedding for both users and items on its own. It embeds both users and items in the same embedding space. It notes which items a particular user likes and also the items that the users with behavior and likings like him/her likes to recommend items to that user. It collects user feedback on different items and uses them for recommendations. Collaborative filtering is further divided into three subcategories: memory-based, model-based, and hybrid collaborative filtering.

Content-based recommendation system: CBRS recommends items based on their features and the similarity between elements of other items. Assuming a user has already seen a movie from the genre of Comedy, CBRS will recommend movies that also belong to the Comedy genre. A content-based recommender works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link). Based on that data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

Python is increasingly being used as a scientific language. Matrix and vector manipulations are extremely important for scientific computations. Both NumPy and Pandas have emerged to be essential libraries for any scientific computation, including machine learning, in python due to their intuitive syntax and high-performance matrix computation capabilities.

NumPy stands for 'Numerical Python' or 'Numeric Python'. It is an open-source module of Python which provides fast mathematical computation on arrays and matrices. Since arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem.

Flask uses the metropolis example engine to dynamically build hypertext markup language pages mistreating acquainted Python ideas like variables, loops, lists, and so on. We've used these templates as a part of this project. Flask may be a small internet framework written in Python. it's classified as a microframework; as a result of it doesn't need explicit tools or libraries. Templates are files that contain static knowledge still as placeholders for dynamic knowledge. An example is rendered with specific knowledge to provide a final document. Flask uses the metropolis example library to render templates that we've employed in the project. The tactic attribute of type|The shape} component tells the net browser a way to send form knowledge to a server. Specifying a price of POST suggests that the browser can send the info to the net server to be processed. This is often necessary once adding knowledge to info, or once submitting sensitive data. within the project POST technique is employed to require needed song name input from the user, then it's processed into the particular cubic centimeter program for recommending the song. In Python, a lambda perform may be a single-line perform declared with no name, which might have any variety of arguments. However it will solely have one expression. Mistreatment associate degree idles perform and together with the specified variables and lambda perform to fetch the prediction of song recommendation from the cubic centimeter engine for Song Recommendation.

II. LITERATURE SURVEY

Personalized Recommender Systems

Personalization issues adapting to the individual desires, interests, and preferences of every user. They're tools for suggesting things to users.

Content-based Recommender Systems

Pasquale Lops, Marco American state Gemmis, and Giovanni Semeraro, 2010 [1] in their paper Content-based Recommender Systems: State of the Art and Trends discusses the most problems associated with the illustration of things, ranging from easy techniques for representing structured information to a lot of complicated techniques returning from {the information|the knowledge|the information} Retrieval analysis space for unstructured data.
This work is split into three components. The primary half presents the essential ideas of content-based recommender systems, a high-level design, and their main blessings and disadvantages. The second half a review of the state of the art of systems adopted in many application domains by describing each classical and advanced technique for representing things and user profiles. The foremost wide adopted techniques for learning user profiles also are conferred. The last half discusses trends and future analysis which could lead towards ensuing generation of systems, by describing the role of User Generated Content as how for taking under consideration evolving vocabularies, and also the challenge of feeding users with lucky recommendations, that's to mention amazingly fascinating things that they could not have otherwise discovered.

Hybrid Recommender Systems

Robin Burke, [2] in his survey Hybrid Recommender Systems: Survey and Experiments, explains numerous recommendation techniques. These techniques show the complementary benefits and downsides. It compares the assorted techniques and shows that techniques area unit higher supported the analysis metrics. This reality has provided an incentive for analysis in hybrid recommender systems that mix techniques for improved performance.
It proposes numerous hybrid approaches which may be accustomed recommendation systems supported the appliance for higher accuracy and results.

Recommendation System Using Association Rules Mining

Luo Zhenghua, 2012 [3] in the realization of individualized recommendation system on book sale applies the association rules in data processing to e-commerce business systems of book sales, styles AN individualized recommendation system of book sales, and introduces the flow of the advice system and therefore the specific realization procedures of information input, knowledge preprocessing, association rules existence and individualized recommendation. Results show that the net website supported this has shown nice performance.

Hybrid Approach for Collaborative Filtering

Gilbert Badaro, Hazem Hajj, Wassim El-Hajj, and Lama Nachman, 2013 [4] in hybrid approach for cooperative filtering for recommender systems talks a couple of new hybrid approach for determining the matter of finding the ratings of unrated things in the user-item ranking matrix by a weighted combination of user primarily based} and item-based cooperative filtering. The projected technique provides enhancements in addressing 2 major challenges of recommender systems: accuracy of recommender systems and scantness of information. The analysis of the system shows the superiority of the answer compared to complete user-based cooperative filtering or item-based cooperative filtering.

The literature survey shows that a hybrid model is projected which mixes user-based cooperative filtering and item-based cooperative filtering by adding the anticipated ratings from every technique and multiplying them with a weight that comes with the accuracy of every technique alone. The approach advantages from the correlation between not solely users alone or things alone however from each at the same time. The analysis was conducted on movielens dataset. the selection of weights was thought of by victimization and adjusting mean absolute error. therefore the survey shows that the hybrid approach improves the information scantness drawback and therefore the accuracy of the system effectively and with efficiency.

Content and collaborative based filtering and association rule mining

Anand Shanker Tewari, Abhay Kumar, and Asim Gopal bartender, [5] proposes a replacement approach to book recommendation system by combining options of content primarily based filtering, cooperative filtering, and association rule mining. The literature survey shows that numerous parameters like content and quality of the book by doing cooperative filtering of ratings by alternative consumers. the aim of this technique is to advocate books to the client that suits their interest. this technique works offline and stores recommendations within the buyer's internet profile. It finds out the class of the book that the client has bought earlier, like a novel, science, engineering, etc. from the consumer's internet profile. It finds out the subcategory of the book.

It performs content-primarily based filtering in class /subcategory, to search out the books that are unit abundant just like the books that the client has bought earlier from the consumer's past history record. On the results of the on top of the step, item primarily based cooperative filtering is performed. This step truly evaluates the standard of the recommending books supported by the rating given to those books by the opposite consumers. From the book dealing info, realize all transactions whose class and subclass are the same as found in step1 and step2.

Non-Personalized Recommender Systems

Non-personalized recommender systems are the only form of recommender systems. They are doing not take into consideration the non-public preferences of the users. The recommendations made by these systems are identical for every client.

Non-Personalized and User-based Collaborative Filtering

Anil Poriya, Neev Patel, Tanvi Bhagat, and Rekha Sharma, Ph. D, [6], in their paper Non-Personalized Recommender Systems and User-based cooperative Recommender Systems describes however websites these days extremely rely on recommender systems. It provides United States insight into 2 common techniques: non customized recommendation and cooperative filtering. Non Customized recommendations use 2 sorts of algorithms: collective opinion recommender and Basic product association recommender.

The literature review describes, collective opinion recommender that essentially recommends restaurants supported the typical score given to that by different customers. The typical is calculated victimization spherical mean ratings. But these averages lack context throughout recommendations. Thus basic product association recommender is employed. It provides helpful non-personalized recommendations in an exceeding context. Recommendations might not be essentially specific to the user however specific to what the user is presently doing (viewing/buying). The recommendations during this system are similar to all or any users and lack personalization and therefore won't attractive to everybody. Thus cooperative filtering is employed. The cooperative recommender systems overcome the dearth of the personalization involved non-personalized recommender systems. Conjointly no item knowledge is required for this approach and its domain freelance. The machine time is low for model primarily based approaches.

Literature Summary

SN	Paper title	Author & Year of Publication	Methodologies Advantages and Disadvantages
1.	Content-based Recommender Systems	Pasquale Lops, Marco de Gemmis and Giovanni Semeraro. 2010	Advantages: Learning of profile is made easy. Quality improves over time. Considers implicit feedback. Disadvantages: Does not completely Overcome the problem of over-specialization and serendipity.
2.	Hybrid Recommender Systems:	Robin Burke 2010	Advantages: The survey shows combine techniques for improved performance. It improves the user preferences for suggesting items to users.
3.	Association rule Mining for recommendation system on the book sale	Luo Zhenghua. 2012	Advantages: The website based on this has shown great performance. Disadvantages: It does not recommend quality content to the users. Does not consider new user cold start problem Not very efficient in terms of performance

4.	Collaborative filtering for recommender systems: User-based and Item-based CF	Gilbert Badaro, Hazem Hajj, Wassim El-Hajj and Lama Nachman. 2013	Advantages: solves the problem of finding the ratings of unrated items in a user-item ranking matrix. It improves the data sparsity problem. Disadvantage: It does not consider the demographic features which would give better results and solve the user cold-start problem.
5.	Content-Based Filtering, Collaborative Filtering, and Association Rule Mining	Anand Shanker Tewari, Abhay Kumar, and Asim Gopal Barman. 2014	Advantages: It considers various parameters like content & quality of the book by doing collaborative filtering of rating of other buyers. It does not have performance problems. It builds the recommendation offline. Disadvantage: It still lacks the new user cold-start problem.
6.	Non-Personalized Recommender Systems and User-based Collaborative Recommender Systems	Anil Poriya, Neev Patel, Tanvi Bhagat, and Rekha Sharma. 2014.	Advantages: The system helps users find items they want to buy from a business. It overcomes the lack of personalization involved with non-personalized recommender systems. It is domain-independent. Disadvantages: The recommendations are not very specific. It still lacks personalization. The computational time is low.

III. Proposed System

This research focuses on determining the most effective DM technique with the highest precision between the different classification techniques to be used. In addition, finding the effect of train/test data ratio on the accuracy of the prediction.

System Architecture

The system architecture is given in Figure 1. Each block is described in this Section.

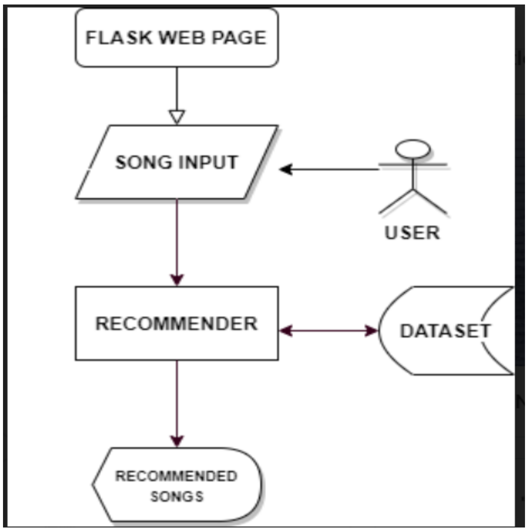


Fig. 1 Proposed System Architecture

A. Data Collection and Understanding Process: The real dataset is used for the research. We have taken music data which contains 2000 records and 15 fields, including categorical and numeric features. Each record in the music data set represents single musical information, and each field in the record represents a feature of that particular employee.

B. Data Preparation and Pre-processing: After the process of data collection is finished, the process of preparing the data is performed. It is important to refine this data so that it can be suitable for the models and generate better results. In this phase we performed tasks like cleaning, filling the missing data, and removing unwanted data. The data of Spotify had various attributes which were not relevant, i.e., was not giving any useful information, like Title, Artist, Top Genre, Energy, BPM, Liveness, etc.; hence these attributes are removed in this phase.

C. Feature Selection: Feature selection is one of the main concepts of DM and Machine Learning. Where it is a process of selecting necessary useful variables in a dataset to improve the results of machine learning and make it more accurate, there are a lot of columns in the predictor variable. So, the correlation coefficient is calculated to see which of them are important and these are then used for training methods. From there, we get the top factors that affect performance.

D. Test and Train Dataset:

Separating data into test datasets and training datasets is an important part of evaluating data mining models as it minimizes the effects of data inconsistency and better understands the characteristics of the model. The test data set contains all the required data for data prediction, and the training data set contains all irrelevant data. We have split the dataset into variable ratios to study the estimation of Prediction.

This paper targets getting the most important variables that may positively affect the accuracy of the features of music performance prediction models using the various feature selection algorithms.

IV. Modeling and Experiments

Before building the model and software infrastructure, the data preprocessing and cleaning step is done, since the function get important features appends all the required rows, there may come NaN values in the dataset which have to be replaced with an empty string.

We can see the most important features selected in Table .

Sr. No	Attributes
1	Title
2	Artist
3	Top Genre

Table 3 Final Attributes used for prediction

The specified features are the appended to form a long string which is later used to find similarity score for each song.

V. Requirement Analysis

A. Software

The operating systems used will be windows 7& above. Programming languages used are Python, HTML5, CSS3, Bootstrap.

B. Hardware

The main memory required is 8 GB & above so that the whole program can reside on the same memory at once. This will avoid the requirement to swap the memory contents of the system. The hard disk drive is required to store the program permanently on the storage. The processor is required to process the data quickly on the system. A Computer/Laptop is required to enable the user to interact with the system while on the go.

VI. Implementation and Result Analysis

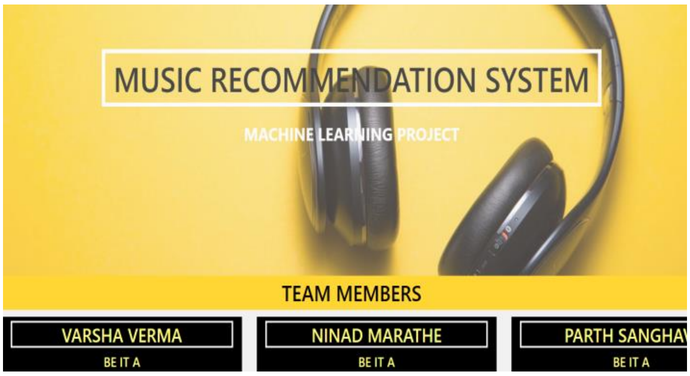
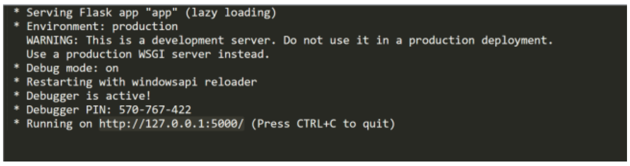
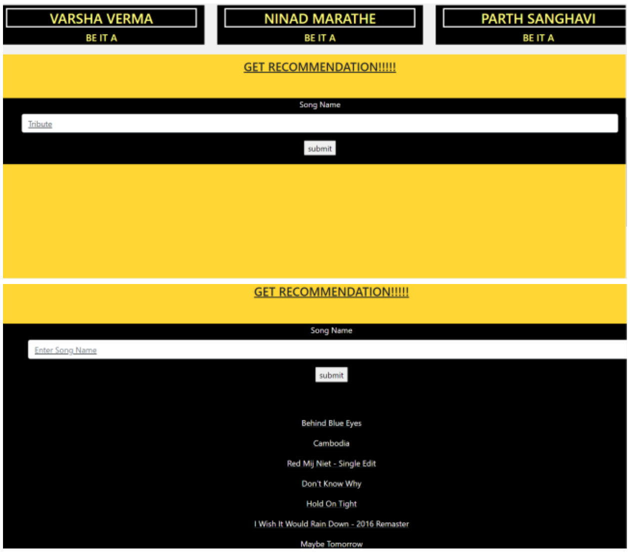


Fig 3 User input page

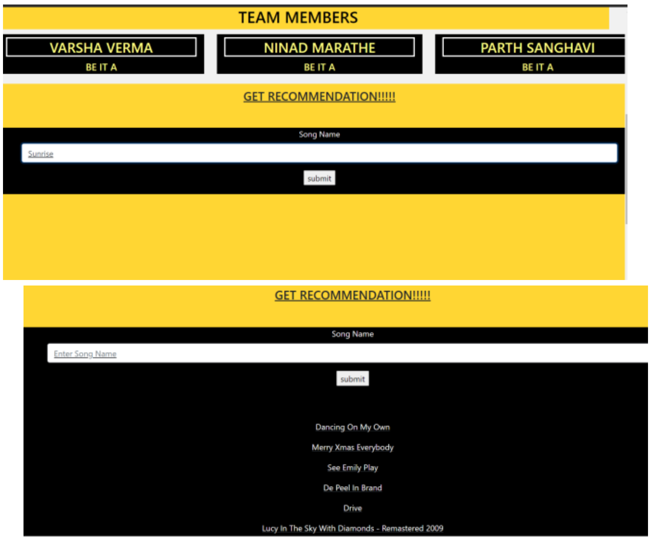


```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with windowsapi reloader
* Debugger is active!
* Debugger PIN: 570-767-422
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

In the system, First user inputs the song which he/she wants; once the required song is inputted by the user, that ten similar songs are recommended to him. Initially, the process takes into consideration by taking three main features, that is Title, Artist, and Top Genre, which is done by taking Angular distance and Euclidean distance. For this, we have taken the class Count Vectorizer and method cosine similarity. Count vectorizer is stored in an object which is used to count the number of terms that appeared in a particular feature; after that, structured data is used by cosine similarity to find the similarity score. Before the data is processed by the count vectorizer class, since we are using multiple parameters/ features to find the similarity score, a function is created to merge the contents of all the rows of the specified features. In case any NaN values are found, they are replaced with an empty string.



Example figure 1.



Example figure 2

Once we get the cosine similarity between the features, we create a list of enumeration for the similarity score. After that, the seven most similar songs are predicted by the model which is presented on the frontend.

V. Conclusion and Future Scope

In the future, we would like to try the following things: 1.Using audio signal (e.g. audio frequency) to recommend songs 2.Trying content-based algorithm 3.Trying Convolutional Neural Network 4.Making the recommender system a real-time system 5, trying clustering techniques to recommend music. Designing a personalized music recommender is complicated, and it is challenging to thoroughly understand the users' needs and meet their requirements. As discussed above, the future research direction will be mainly focused on user-centric music recommender systems. A survey among athletes showed practitioners in sport and exercise environments tend to select music in a rather arbitrary manner without full consideration of its motivational characteristics. Therefore, future music recommenders should be able to lead the users to reasonably choose music. In the end, we are hoping that through this study, we can build the bridge among isolated research in all the other disciplines.

VI. Acknowledgment

It is our privilege to express our sincerest regards to our supervisor Dr. Prashant Nitnaware for the valuable inputs, able guidance, encouragement, whole-hearted cooperation, and constructive criticism throughout the duration of this work. We deeply express our sincere thanks to our Head of the Department, Dr. Sharvari Govilkar, and our Principal, Dr. Sandeep M. Joshi, for encouraging and allowing us to present this work.

VII. REFERENCES

[1]. Luo Zhenghua, "Realization of Individualized Recommendation System on Books Sale," IEEE 2012 International Conference on Management of e-Commerce and e-Government. pp.10-13.

[2]. Tewari, A.S. Kumar, and Barman, A.G, "Book recommendation system based on combining features of content-based filtering, collaborative filtering and association rule mining," International Advance Computing Conference (IACC), IEEE, pp 500 – 503, April 2014.

[3]. Robin Burke, "Hybrid Recommender Systems: Survey and Experiments", California State University, Department of Information Systems and Decision Sciences, Vol. 12, No. 4, pp. 331-370, March 2012.

[4]. Anil Poriya, Neev Patel, Tanvi Bhagat, and RekhaSharma, "Non-Personalized Recommender SystemsandUser-basedCollaborative Recommender Systems", International Journal of Applied Information Systems (IJAIS), FCS, Vol. 6, No. 9, March 2014.

[5]. Fang, J., Grunberg, D., Luit, S., & Wang, Y. (2017, December). Development of a music recommendation system for motivating exercise. In Orange Technologies (ICOT), 2017 International Conference on (pp. 83-86). IEEE.

[6]. Nakamura, K., Fujisawa, T., & Kyoudou, T. (2017, October). Music recommendation system using lyric network. In Consumer Electronics (GCCE), 2017 IEEE 6th Global Conference on (pp. 1-2). IEEE.

[7]. Keita Nakamura, Takako Fujisawa. Music recommendation system using lyric network, Journal of 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), 2017

[8]. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-7, May, 2019

[9]. 4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12–13 September 2019

[10]. Arditi D. Digital Subscriptions: The Unending Consumption of Music in the Digital Era. Popular Music and Society. 2017

Cite this article as :

Varsha Verma, Ninad Marathe, Parth Sanghavi, Dr. Prashant Nitnaware , "Music Recommendation System Using Machine Learning ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7 Issue 6, pp. 80-88, November-December 2021. Available at
doi : https://doi.org/10.32628/CSEIT217615
Journal URL : https://ijsrcseit.com/CSEIT217615