RV College of Engineering®

(Autonomous Institution Affiliated to VTU, Belagavi)

OBJECT ORIENTED PROGRAMMING SYSTEM(OOPs)
PROJECT SYNOPSIS

AUTOMATED ATTENDANCE SYSTEM USING FACE RECOGNITION

Submitted by

DILEEP SHARMA	**1RV20IS017**
KARTHIK PAI	**1RV20IS020**

INFORMATION SCIENCE AND ENGINEERING

Under the guidance of

Sushmita N

Assistant Professor

1. Introduction

Recommender systems are the systems that are designed to recommend things to the user based on many different factors. These systems predict the most likely product that the users are most likely to purchase and are of interest to. Companies like Netflix, Amazon, etc. use recommender systems to help their users to identify the correct product or movies for them.
Music is one of the popular entertainment media in the digital era. Music can be categorized into several genres, such as pop, rock, jazz, blues, folk etc. Listening to music in the digital age is easier because of the features on the smartphone that can play music offline and online. Nowadays, the availability of digital music is very abundant compared to the previous era, so to sort 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 music libraries automatically and suggest songs that are suitable for users. Music streaming applications like Spotify and Pandora have features to recommend music to users.

2. Problem statement

To build a simple music recommendation system

Music Recommendation system is a very specific type of recommendation system which as the name suggests recommends music to its users to provide an overall good experience to the user. Building a solid system will help in keeping the customer retention and will attract a new set of users. This will help us in having a solid user base which will in turn provide us with a lot of data. This data can be used to further improve our recommendation system and the cycle begins once again. Our recommendation system should also have a pretty easy to use UI which would make the usage of the app to the customers easy.

3. Objectives

a. Implementing mainly 2 types of recommendation

    - Popularity based recommendation system
    - User based recommendation system

b. Using cosine similarities and possibly an ML model to get the accurate recommendation on our Dataset
c. Implement a basic search engine based on string matching
d. Implementing a basic UX/UI using JavaFX and OOPs concepts to showcase our recommendation system

4. Methodology

i) Implementation of recommendation system

Any recommendation system can recommend based on many recommendation systems. But we here are using mainly 2 types of recommendation systems.

- Popularity based recommendation system: Based on the features like no of plays and duration of plays we would assign a popularity value for each song on a suitable scale later on the basis of this feature we filter out the top most songs to recommend for the user first time entering into the music player. Here basic data filtration can be done by using a java library called Weka which is popular for doing these kind of works
- User based recommendation system: As a user starts playing songs of his interest, we would run a function mainly based on cosine similarities and search for songs which are most similar to the song played and then recommend it to the user. Later we would also try to implement a function which also prioritizes songs based on the other users playing a song right after the song the user is playing basically following the rule like-minded people like same kind of songs.

ii) Using cosine similarities and appropriate ML model

Cosine similarity is a metric, helpful in determining, how similar the data objects are irrespective of their size. In cosine similarity fall numerical features of each object of a dataset are treated as vectors. The formula to compute the cosine similarity of 2 vectors is

```
Cos(x, y) = x . y / ||x|| * ||y||
```

The dissimilarity between the two vectors ‘x’ and ‘y’ is given by

```
Dis(x, y) = 1 - Cos(x, y)
```

Our ML model should be able to recommend two main sections to the user the “For you section” and “Upcoming songs section”. Basic ML concepts such as “User based collaborative filtering” and “Item based collaborative filtering” can be used to achieve the same.

iii) Implement a basic search engine based on string matching

A music application must have a suitable search engine. Our search engine must have features such as string matching, filtered based searching etc. Robin Karp’s Algo can be used for string matching. Weka for filter-based searching

iv) Implementing a basic UX/UX using java FX and OOPs concepts to showcase our recommendation system

To build any real-world application we always require concepts of OOPs to be properly used. Coming to our scenario we basically require 2 Major classes namely RecommendBasedOnPopularity and RecommendBasedOnUser which further will be inheriting the interface Recommender. Recommender interface should be declaring all the methods that need to be implemented by classes which implement it. The methods mainly include filtering songs based on different features such as languages, popularity, genre, to get the songs list based on what user types in the search bar using string matching etc. . JavaFX should be used to provide a satisfactory user experience and make it easy to use.

5. SRS

Technology used

- Weka
- Maven
- Java FX

System Requirements

- OS: Windows 10 or Linux 20.04 having JRE

6. Conclusion

In this way we can build a pretty solid recommendation system. Both popularities based and user-based recommendations are used to enrich user experience. Having a search engine along with a pretty good UI/UX will help provide an overall satisfactory time for the users and also this would enhance our knowledge about OOPs concept and also JavaFX and we will be knowing the power of Java.