# SYMPHONY: As You Like It!
## (Music Recommendation System)

**Capstone Project Report**

**End Semester Evaluation**

**Submitted by:**

**(101503004) ABHIMANYU SHARMA**

**(101503023) AMANDEEP SINGH**

**(101503086) HARNOOR SINGH BEDI**

**BE Fourth Year, Computer Engineering**

**CPG No: 142**

Under the Mentorship of

Ms. Vineeta Bassi (Assistant Professor)

Mr. Antriksh Goswami (Lecturer)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

**Computer Science and Engineering Department**

**TIET, Patiala**

**November 2018**

# ABSTRACT

This project involves the use of data analysis methods for the development of a Music Recommendation System. The user would be able to input his song to the system and the Recommendation System would suggest similar songs. The songs are compared based on technical, "inner" values - their loudness, their tempo, their mode for example. Hence the team aims to build a Content based Recommendation System which recommends songs on the basis of similarity of features. The Million Song Dataset from Kaggle is used to implement data analysis methods. The dataset contains metadata of one million songs and is the most comprehensive music dataset available.

# DECLARATION

We hereby declare that the design principles and working prototype model of the project entitled Symphony: As You Like It! (Music Recommendation System) is an authentic record of our own work carried out in the Computer Science and Engineering Department, TIET, Patiala, under the guidance of Ms. Vineeta Bassi and Mr. Antriksh Goswami during 7th semester (2018).

Date:

| Roll No. | Name | Signature |
|---|---|---|
| 101503004 | Abhimanyu Sharma | |
| 101503023 | Amandeep Singh | |
| 101503086 | Harnoor Singh Bedi | |

*Counter Signed by:*

Faculty Mentor:

Ms. ____________________________
Assistant Professor
Computer Science & Engineering Dept.
TIET, Patiala

Mr. ____________________________
Lecturer
Computer Science & Engineering Dept.
TIET, Patiala

# ACKNOWLEDGEMENT

We would like to express our thanks to our mentors Ms. Vineeta Bassi and Mr. Antriksh Goswami. They have been of great help in our venture, and an indispensable resource of technical knowledge. They are truly amazing mentors to have.

We are also thankful to Dr. Maninder Singh, Head, Computer Science and Engineering Department, entire faculty and staff of Computer Science and Engineering Department, and also our friends who devoted their valuable time and helped us in all possible ways towards successful completion of this project. We thank all those who have contributed either directly or indirectly towards this project.

Lastly, we would also like to thank our families for their unyielding love and encouragement. They always wanted the best for us and we admire their determination and sacrifice.

Date:

| Roll No. | Name | Signature |
|---|---|---|
| 101503004 | Abhimanyu Sharma | |
| 101503023 | Amandeep Singh | |
| 101503086 | Harnoor Singh Bedi | |

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **MICE** | Multiple Imputation by Chained Equations |

# INTRODUCTION

## 1.1 Project Overview

Rapid development of mobile devices and internet has made possible for us to access different music resources freely. The number of songs available exceeds the listening capacity of single individual. People sometimes feel difficult to choose from millions of songs. Moreover, music service providers need an efficient way to manage songs and help their customers to discover music by giving quality recommendation. Thus, there is a strong need of a good recommendation system. Currently, there are many music streaming services, like Pandora, Spotify, etc. which are working on building high-precision commercial music recommendation systems. These companies generate revenue by helping their customers discover relevant music and charging them for the quality of their recommendation service. Thus, there is a strong thriving market for good music recommendation systems.

The explosive growth in the amount of available digital information and the number of visitors to the Internet have created a potential challenge of information overload which hinders timely access to items of interest on the internet. Information retrieval systems, such as Google have partially solved this problem, but prioritization and personalization of information were absent. This has increased the demand for recommender systems more than ever before. Recommender systems are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of large amount of dynamically generated information according to user’s preferences, interest, or observed behavior about item. Recommender system could predict whether a particular user would prefer an item or not based on the user’s profile.

The project involves implementation of a Content Based Music Recommendation System. The aim is to implement k-means clustering algorithms to group together similar songs based on their features. The Million Song dataset [1] provided by Kaggle will be used to implement the data analysis methods.

## 1.2 Need Analysis

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. Facing a massive collection of music, users are unable to make a decision and have no idea of what to listen to. The users sometimes have problems discovering new songs when using music streaming websites. They wish the streaming websites to provide recommendations for them. Even according to the CEO and founder of Spotify, Daniel Ek, users have frequently voiced their desire of finding new music to listen to. Obviously, a music recommender system is essential in music streaming websites. Users demand an effective music recommender system because music streaming websites offer numerous items to choose from within a limited period of time which is insufficient to evaluate all possible options. With the help of a recommender system, users can skip over the information overload and get customized recommendations from the system.

## 1.3 Research Gaps

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 [4] provides data which is open and large scale which facilitates academic research in user centric music recommender system which hasn't been studied a lot.

## 1.4 Problem definition and scope

The central problem that the project aims to solve is the recommendation of songs to a user. The recommendation system suggests similar songs to the song entered by the user in the query. So the question is "How does the system provide recommendations?". In order to discern whether songs are similar to those in a query, our methods use the metadata collected from a song to weigh its similarity with another song. Hence the most relevant songs are recommended to the user.

## 1.5 Assumptions and constraints

The Internet connection is a constraint for the application. Since the application fetches data from the server over the Internet, it is crucial that there is an Internet connection for the application to function. Another assumption is that the user has a web browser and a capable hardware in order to launch the website.

Table 1: Assumptions and Constraints

| S. No. | Assumptions |
|---|---|
| **1** | This is a Music Recommendation system and it is used in the following application:<br>• To provide recommendations to the user of similar music as entered in the query<br>• Music is compared on the basis of technical values such as Loudness, Tempo and Mode.<br>Assuming that the user will enter the song id into the query after searching for his song in the dataset. |
| **2** | It is assumed that the Million Song Dataset from Kaggle will be used for the project. It consists of metadata of one million songs and it will be used for providing the demo for the presentations. It is assumed that the user is familiar with an internet browser and has basic computer proficiency. Since the application is a web based application there is a |

| | |
|---|---|
| | need for the internet browser. It will be assumed that the users will possess decent internet connectivity. |
| **3** | One assumption about the product is that it will always be used on a system that has enough performance. Since the application works on a very large dataset, the presence of a high performance system is of utmost importance or else the application may not work as intended. |

## 1.6 Approved objectives

- Build a Music recommendation system with the goal of predicting the songs that a user is going to listen.

- Integrate this system into a webpage where user can search for their music and get recommendations according to their taste.

- To improve overall efficiency of the recommendation system using more optimized algorithms.

## 1.7 Methodology used

*Data Set*:
Data is provided by Million Song Data Challenge hosted by Kaggle. It was released by Columbia University Laboratory for the Recognition and Organization of Speech and Audio. It contains metadata of one million songs.

*Algorithm*:
The algorithm used to recommend songs is the K-means clustering algorithm. It is an unsupervised learning algorithm. It clusters together songs having similar features. The features that we chose are loudness, tempo and mode. The songs within a cluster

are those ones with maximum similarity. The user enters a song that he wants recommendations for and the system recommends the most similar songs. It is a content based recommendation system which works on the principle of item-item similarity.

## 1.8 Project outcomes and deliverables

- At the end of this project, the system will be able to provide recommendations to the user based on the song entered in the user query.

- With the help of our system, users will be able to get music recommendations according to their taste through a fully functional web application.

- The system will save the time which the user spends on manually searching for the music that he may like.

## 1.9 Novelty of work

The existing content based music recommendation systems recommend music from similar artists and genres. The existing collaborative filtering music recommendation systems can recommend music based on what other users like that also like your song. Our system is bit different from the existing systems because it recommends songs by comparing technical, "inner" values - their loudness, their tempo, their mode for example.

## 2.1 Literature survey

### 2.1.1 Theory associated with problem area

It is obvious that music plays an important role in many people's daily life. No matter if one is a huge fan of music or just randomly listens to music for fun, it cannot be denied that music is a major entertainment factor. From vinyl records to cassette tapes, from CD to mp3, the ways of listening to music have changed. With the help of technology, music can be enjoyed in a more and more convenient way. Nowadays, with the rapid development of the Internet, it is getting common to use music streaming services. Compared to other ways of providing music, streaming websites can provide more and better services. There are a lot of advantages of using music streaming websites: customers pay less to listen to music than with iTunes or real CDs; the number of music collections in streaming websites is huge; it is much more convenient to listen to music online etc.

According to Karp (2014) [8], in the first half of 2014, the number of downloads of singles and albums dropped by 11% and 14%, whereas the number of users of streaming services increased by 28%; these figures make it obvious that more and more people have changed their ways of listening to music. Streaming music services started to change people's habits of listening to music. There are already several music streaming websites, for instance: Spotify, Beasts music, Pandora. Some big companies have started their own music recommendation services, for example: Google play music, Sony music unlimited, X-box music etc. The user group of streaming services is gigantic in number. Using the music streaming services can represent an innovative and superior experience for the user. One important reason why more and more people choose to use music streaming services is that they thus can build up a massive music collection. However, this advantage also entails a problem: information overload. This problem becomes obvious on streaming websites. Facing a massive collection of music, users are unable to make a decision and have no idea of what to listen to. The users sometimes have problems discovering new songs when using music streaming websites. They wish the streaming websites to provide recommendations for them. Even according to the CEO and founder of Spotify [12], Daniel Ek, users have frequently voiced their desire of finding new music to listen to. Obviously, a music recommender system is essential in music streaming websites. Users demand an effective music recommender system because music streaming websites offer numerous items to choose from within a limited period of time which is insufficient to evaluate all possible options. With the help of a recommender system, users can skip over the information overload and get customized recommendations from the system.

### 2.1.2 Existing systems and solutions

To meet users' demands for a recommender system, there are some music streaming websites already providing music recommendation services, for example: Spotify [13], Pandora, Beats music etc. The ways how they compile their recommendation lists varies between Companies. Some websites make up recommendations based on users' listening records; some recommend the music that the "neighbour user" listens to, which means that the system assumes that they share a similar taste, and other websites recommend music based on user's mood. Although there are already lots of different ways to draw up recommendations, users are still not satisfied with the recommendation service.

### 2.1.3 Research Findings for Existing Literature

Table 2: Research findings for existing literature

| S. No. | Roll Number | Name | Paper Title | Tools/ Technology | Findings | Citation |
|---|---|---|---|---|---|---|
| 1 | 101503004 | Abhimanyu Sharma | Preliminary study on a recommender system for the million songs dataset challenge | Recommender Systems | Learnt about Collaborative filtering | Aiolli. [4] |
| 2 | 101503004 | Abhimanyu Sharma | Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks | Neural networks | Cold start problem | Liang *et. al.* [3] |
| 3 | 101503004 | Abhimanyu Sharma | Content-based, Collaborative recommendation | Content based recommendation systems | Combination of content based and collaborative systems eliminates weakness of both. | Balabanovic [11] |
| 4 | 101503004 | Abhimanyu Sharma | Hybrid web recommender systems | Hybrid recommender systems | Hybrid systems have better | Burke [19] |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | performan ce | |
| 5 | 101503004 | Abhimanyu Sharma | Multiple Imputation by Chained Equations: What is it and how does it work? | MICE Imputation technique | Predicting missing values in dataset. | Azur *et. al.* [17] |
| 6 | 101503004 | Abhimanyu Sharma | Normalisation: A Preprocessing stage | Dataset normalisation and scaling | Learnt importanc e of data preprocess ing | Patro and Sahu. [21] |
| 7 | 101503023 | Amandeep Singh | Collaborative filtering recommender systems | Recommendor System algorithms | Collaborat ive filtering technique | Ekstra nd *et. al.* [15] . |
| 8 | 101503023 | Amandeep Singh | Efficient top-n recommendations for large scale binary rated datasets. | Memory based collaborative filtering | Implemen tation of algorithms on dataset | Aiolli [5] |
| 9 | 101503023 | Amandeep Singh | Probablistic models for unified collaborative and content based recommendation in Sparse-data environments | Probablistic models | Unified content based and collaborati ve systems. | Antho ny *et. al.* [7] |
| 10 | 101503023 | Amandeep Singh | Towards the Next Generation of Recommedor Systems: A | Rating estimation methods | Extension s to recommen | Adoma vicius and |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | Survey of the State of the Art and Possible Extensions | | der systems | Tuzhilin [6] |
| 11 | 101503023 | Amandeep Singh | An Efficient k-Means Clustering Algorithm: Analysis and Implementation | k-Means Clustering Algorithm | Learnt theory behind k-Means Clustering Algorithm | Kanungo *et. al.* [23] |
| 12 | 101503023 | Amandeep Singh | A Scalable, Accurate Hybrid Recommender System | Hybrid Recommender Systems | Hybrid recommender systems are best suited for large scale systems. | Ghazanfar, and Bennett [14] |
| 13 | 101503086 | Harnoor Singh Bedi | The million-song dataset challenge | Music information retrieval | Learnt about the Million Song dataset. | McFee *et. al.* [1] |
| 14 | 101503086 | Harnoor Singh Bedi | Content- Based Recommendation Systems | Recommendation by content based systems | Recommendation on basis of item-item similarity | Pazzani, and Billsus [16] |
| 15 | 101503086 | Harnoor Singh Bedi | Hybrid Recommendor Systems: Survey and Experiments | Hybrid recommendation systems | Hybrid systems mask weakness of | Burke [18] |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | individual systems | |
| 16 | 101503086 | Harnoor Singh Bedi | Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. | Graph Theory | Modern approaches in collaborative filtering. | Aggarwal *et. al.* [2] |
| 17 | 101503086 | Harnoor Singh Bedi | Algorithm AS 136: A K-means Clustering Algorithm | K-Means clustering algorithm | K-Means algorithm use in unsupervised learning. | Hartigan and Wong [10] |
| 18 | 101503086 | Harnoor Singh Bedi | Being accurate is not enough: how accuracy metrics have hurt recommender systems | Knowledge based recommender systems. | Need to move beyond conventional accuracy metrics in recommender systems | McNee *et. al.* [22] |

### 2.1.4 The Problem That Has Been Identified

According to Karp (2014) [9] the music recommended by the recommender systems in music streaming websites does not match with the user's taste. Sometimes the music which is recommended was completely different from what they like. A music recommender system is a however, is supposed to provide good recommendations for users to solve the information overload problem. However, it has become obvious that the music recommender systems do not meet the demands of the users. The question is: What causes this problem? [24] There must be some drawbacks existing in the current music recommender systems. Any basic music recommender system consists of several different components, such as:

- The way of drawing up recommendations: whether the system compiles the recommendations based on data of users' behaviors or users' mood or the "neighbour user's" taste.

- The interface design: whether it is easy for users to understand and apply.

- The feedback system: whether it can actually support the recommender system to get feedback from users and in this way to improve the service. Drawbacks in any part of the recommender system may lead to the "un-customized" problem: the recommendations provided by the system are not tailored to users' demand.

### 2.1.5 Survey of Tools and Technologies Used

The current music recommendation systems use the following two algorithms:

*Content Based Recommendation:*

Content-based recommendation [20] is an inheritance and development of information filtering technology. It is based on the content of user profiles to provide a recommendation service without the user's evaluation. Content-based recommendation uses machine language to a acquire information on the user's interest by relating to the content of the user profile. Use characterization methods, content-based recommendation can offer some choices to the use and then get the user's feedback in content-based recommender systems, the items or those objects are defined by characteristics and related attributes. Content-based recommendation approaches predict the user's interest by using text only, users' ratings are not involved are during the process of the prediction. The user data model depends on the learning method. Decision trees, neural networks and vector-based representation methods are the commonly used. The user data model perhaps varies from user to user. The advantages of the content- based recommendation approach are summarized as follows:

- Other user information is not required in the recommendation process, it is easier to provide the recommendation service at the initial stage of the system
- It can provide recommendation service to users with special interests
- It is able to recommend new or "not mainstream" items
- It can list the recommended items by content characteristics.

*Collaborative Filtering*

Collaborative filtering [25] recommendation is one of the earliest applied techniques and it has successfully spread and entered the recommender system field. It generally uses the "K-nearest neighbor (KNN)" technique which is based on the user's historical records. The music taste of users calculates the distances between the different users. Collaborative filtering recommendation uses the target "user's nearest neighbor user" to weight and evaluate the value of the product. Collaborative filtering recommendation predicts the extent of user's preference for a specific target product. Based on the usage data and the user's interests, the system searches the "neighbour users" [15] who share similar interests with the user. Then the recommendation system recommends the contents that "neighbor users" are interested in to the user. The general idea of how collaborative recommender systems work is easy to understand. In other words, this approach is also commonly accepted in daily life, people refer to friends' recommendations when making their own decisions. Collaborative filtering has been frequently used in e-commerce recommender systems during the past few years. Collaborative filtering provides recommendations for the target users based on other users' evaluation of content.

## 2.2 Standards

**WEB 2.0**

Web 2.0, refers to World Wide Web websites that emphasize user-generated content, usability participatory culture and interoperability (this means that a website can work well with other products, systems, and devices) for end users. The term was invented by Darcy Di Nucci in 1999 and popularized several years later by Tim O'Reilly and Dale Dougherty at the O'Reilly Media Web 2.0 Conference in late 2004. Web 2.0 does not refer to an update to any technical specification, but to changes in the way Web pages are designed and used. The transition was progressive and there is no precise date on which the change occurred.

## 2.3 Software Requirements Specification

### 2.3.1 Introduction

2.3.1.1 Purpose

The purpose of this project is to provide song recommendations to the user. The recommendation system aims to recommend similar songs as the ones entered by the user in their query. The recommendations are generated on the basis of similarity between the songs.

#### 2.3.1.2 Intended Audience and Reading Suggestions

The application software made by us is easily adapted by the audience as it is very easy to use by a person possessing basic computer knowledge. The user of the software just needs the basic knowledge of surfing websites in order to use the system to get recommendations.

### 2.3.2 Overall Description

#### 2.3.2.1 Product Perspective

The product is supposed to be an open source system. It is a web based system implementing client-server model. It composed of presentation, application, data access and security layers which would provide the platform to house the applications. These portals and existing applications will share data using web services.

#### 2.3.2.2 Product Features

The product comprises of following features-:

- Search functionality through which user can search for songs in dataset.
- Recommendation algorithm running efficiently.
- System generates recommendations which are shown to the user through a web interface.

### 2.3.3 External Interface Requirements

#### 2.3.3.1 User Interfaces

Web based Graphical User Interface (GUI) will be provided. It will be ensured that the interface is user friendly. Interface will be designed so that with minimum number of clicks user should be able to access desired information. Screens will be ergonomically designed. Wherever possible, input fields will be pre-populated.

2.3.3.2 Software Interfaces

User will be able to access the Recommendation system using web browser on their system. The Recommendation system software is written in Python language.

#### 2.3.4 Other Non-functional Requirements

Since we will give the priority to the accuracy of the software, the performance of the Music Recommender will be based on its accuracy on recommendations.

The system should generate and provide recommendations to the user in reasonable time.

## 2.4 Cost Analysis

The development tools that we will use for this project are open source. We will use Python for our project. Since it is open source, we will not incur any cost for the development of this project. The dataset is freely available. In addition to this if we want to make our project live, there will be cost of servers and domain names. Domain names cost around 1000-2000 rupees per year. Servers will cost around 700-800 rupees per year.

.

# METHODOLOGY ADOPTED

## 3.1 Investigation technique

Investigative Technique Involved: COMPARATIVE.

Our team needed to select an algorithm to recommend music in this project. To do this, comparative investigation technique was used in which a comparison was made between various recommendation algorithms and the one which was most suitable for our project was selected.

Our team studied about some existing algorithms which are used for music recommendation. We weighed the pros and cons of the existing algorithms. Some algorithms were implemented by our team and finally it was decided that a content based system using k-means clustering algorithms should be implemented. This algorithm was chosen due to its ease of implementation and good performance on our test system.

## 3.2 Proposed Solution

- We present a Recommendation System which will aid the user in getting song recommendations according to his taste.
- We aim to reduce the User's time in searching for new music that he may like.
- The Recommendation System will be integrated into an easy to use web application which anyone with basic computer proficiency can operate.

## 3.3 Work Breakdown Structure

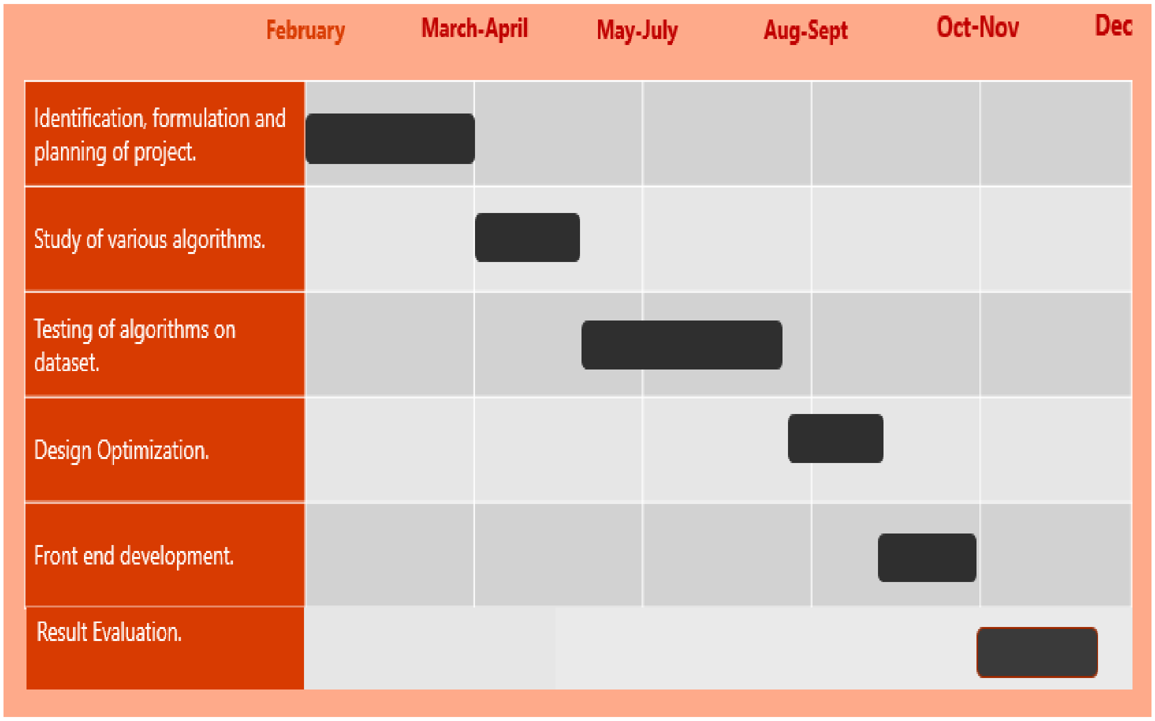



Figure 1: Work Breakdown Structure

## 3.4 Tools and Technologies Used

- The recommendation algorithms are implemented in Python Language.
- The web application is developed using Flask web framework.
- Scikit-learn, NumPy and h5py packages are used.

## 4.1 Block diagram

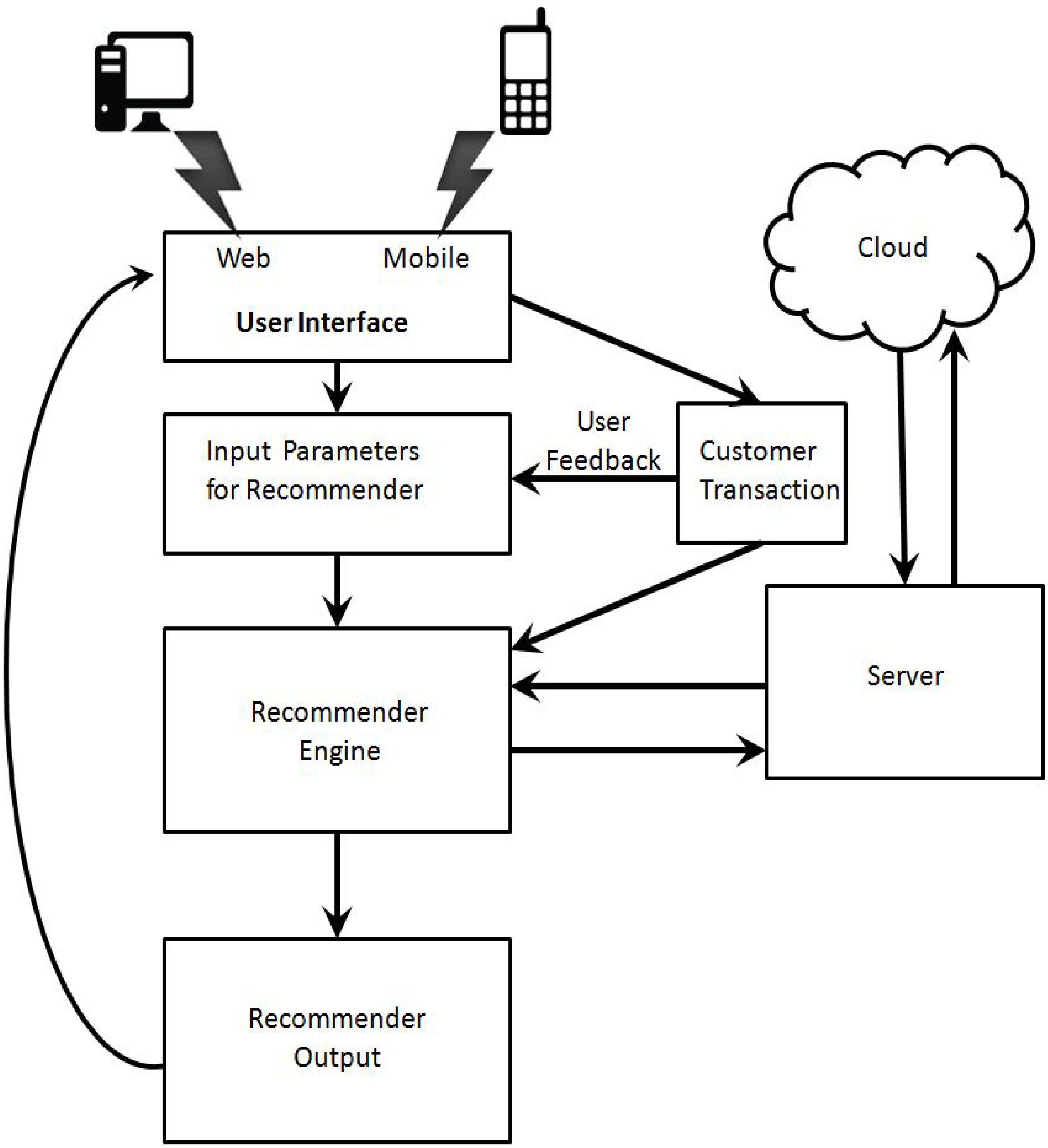



Figure 2: Block diagram

## 4.2 Architecture of recommendation system

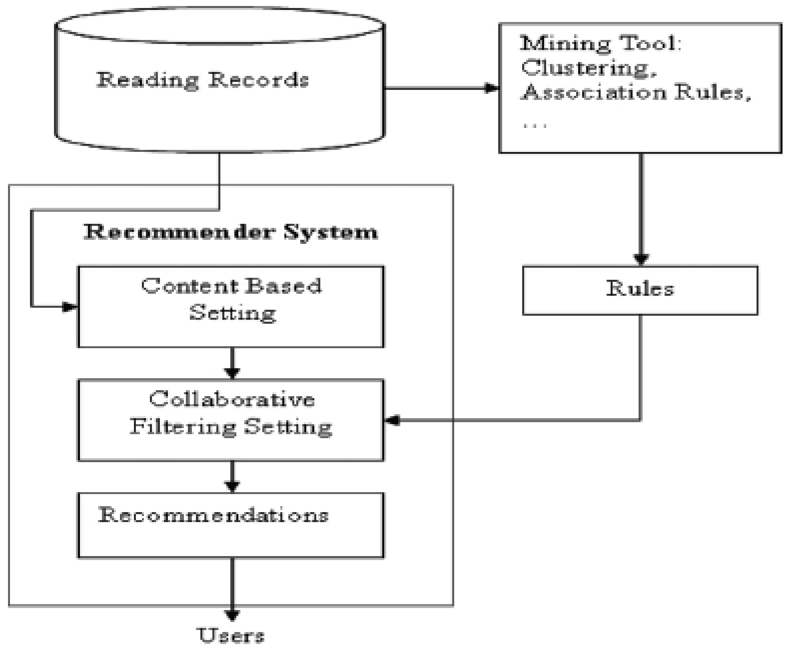



Figure 3: Architecture of recommendation system

# IMPLEMENTATION AND EXPERIMENTATION

## 5.1 Experimental Setup

In this project we aim to build a music recommendation system which recommends music to the user according to his taste. The songs are compared on the basis of their technical values like loudness, tempo, pitch etc. and similar songs are recommended to the user. For experimentation purposes, the recommender system runs on localhost in Firefox web browser.

## 5.2 Experimental Analysis

### 5.2.1 Data

The dataset used for this project is the Million Song Dataset. It is provided by Kaggle. The data has been compiled by Columbia University, New York. It contains metadata of one million songs. It is the most comprehensive music database currently available.

### 5.2.2 Performance Parameters

The performance of a recommendation system is judged by the response time for generating recommendations and the accuracy of the generated recommendations.

## 5.3 Testing Process

### 5.3.1 Test plan

5.3.1.1 Features to be tested

For our music recommender system, we aim to test two features:

- Search feature
- Generate recommendations feature

#### 5.3.1.2 Test strategy

First we aim to test the search feature. Our system provides the user the ability to search based on the name of artist or the name of the song. So we test it by providing various search queries. We also search for an artist which is not present in the database. Secondly we aim to test is the recommendation feature. To test that we provide various queries and see the results. We test for various amounts of recommended songs (by default 10). If the user does not provide a song for recommendation, then the system randomly selects a song from the database and provides recommendations for that song.

### 5.3.2 Test cases

1. Search feature

   1.1 Artist in database. e.g.: Coldplay.

   1.2 Artist not in database. e.g.: aabbccc

2. Recommendations feature

   2.1 Various different amount of recommended songs. e.g. default, 20, 30.

   2.2 Random selection of song.

### 5.3.3 Test results

1. Search feature

   1.1 Artist in database

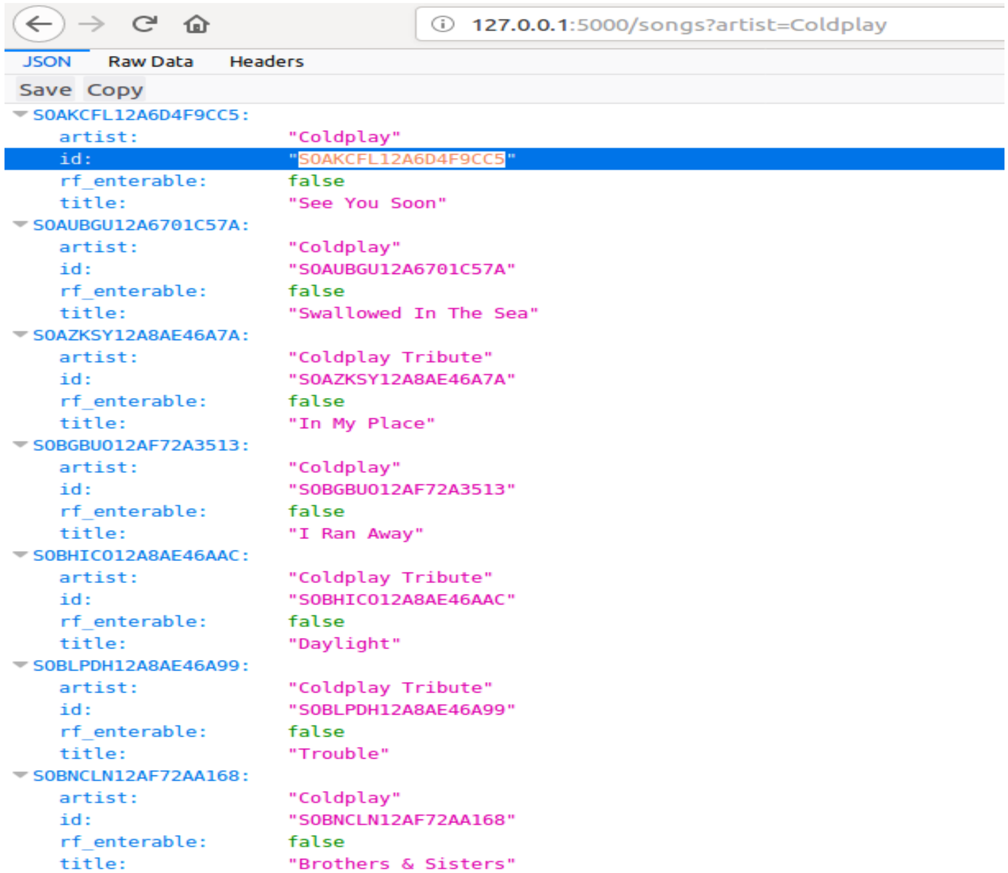



Figure 4: Artist in database

## 1.2 Artist not in database

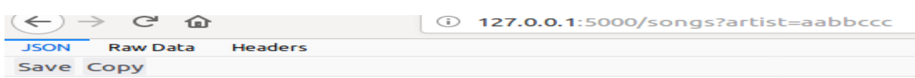



Figure 5: Artist not in database

2. Recommendation feature

## 2.1 Default number of recommended songs


Figure 6: Default number of recommended songs

## 2.2 Varying number of recommended songs

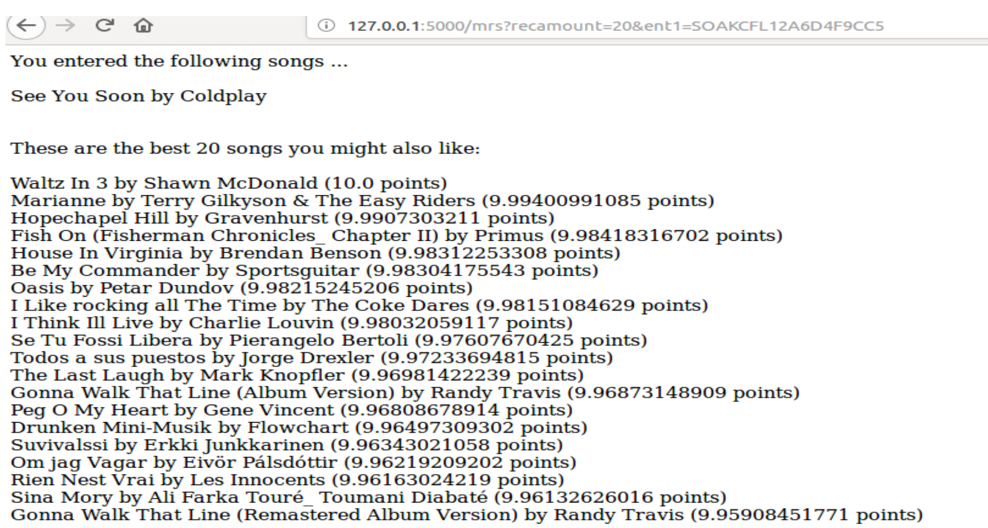
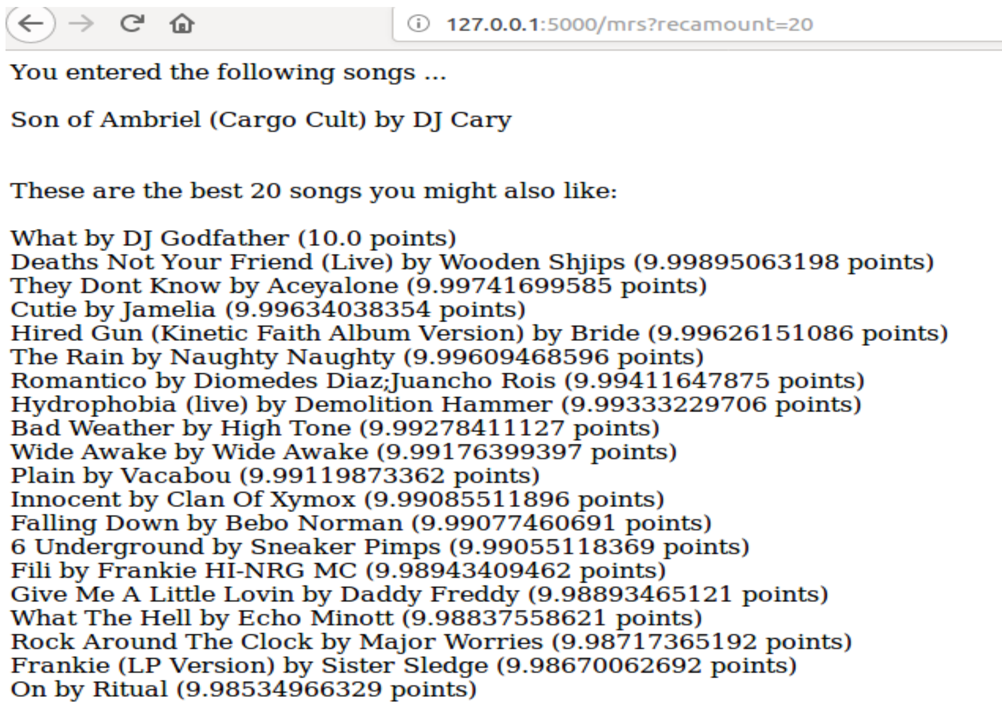



Figure 7: Varying number of recommended songs

2.3 Random song selection


127.0.0.1:5000/mrs?recamount=20

You entered the following songs ...

Son of Ambriel (Cargo Cult) by DJ Cary

These are the best 20 songs you might also like:

What by DJ Godfather (10.0 points)
Deaths Not Your Friend (Live) by Wooden Shjips (9.99895063198 points)
They Dont Know by Aceyalone (9.99741699585 points)
Cutie by Jamelia (9.99634038354 points)
Hired Gun (Kinetic Faith Album Version) by Bride (9.99626151086 points)
The Rain by Naughty Naughty (9.99609468596 points)
Romantico by Diomedes Diaz;Juancho Rois (9.99411647875 points)
Hydrophobia (live) by Demolition Hammer (9.99333229706 points)
Bad Weather by High Tone (9.99278411127 points)
Wide Awake by Wide Awake (9.99176399397 points)
Plain by Vacabou (9.99119873362 points)
Innocent by Clan Of Xymox (9.99085511896 points)
Falling Down by Bebo Norman (9.99077460691 points)
6 Underground by Sneaker Pimps (9.99055118369 points)
Fili by Frankie HI-NRG MC (9.98943409462 points)
Give Me A Little Lovin by Daddy Freddy (9.98893465121 points)
What The Hell by Echo Minott (9.98837558621 points)
Rock Around The Clock by Major Worries (9.98717365192 points)
Frankie (LP Version) by Sister Sledge (9.98670062692 points)
On by Ritual (9.98534966329 points)

Figure 8: Random song selection

## 5.4 Inferences drawn

In all of the test cases specified above, our project works perfectly fine. The search feature provides the required search results to the user. It returns nothing in case the searched artist is not present in the database. The recommendation feature also works fine. It provides the required amount of recommendations to the user. In case the user does not enter a song, it selects a random song from the database and provides recommendations for that song.

## 5.5 Validation of objectives

Table 3: Validation of Objectives

| S.No. | Objective | Status |
|---|---|---|
| 1 | Build a Music recommendation system with the goal of predicting the songs that a user is going to listen. | Successful |
| 2 | Integrate this system into a webpage where user can search for their music and get recommendations according to their taste. | Successful |
| 3 | To improve overall efficiency of the recommendation system using more optimized algorithms. | Successful |

# CONCLUSIONS AND FUTURE DIRECTIONS

## 6.1 Conclusion

In this project we have built a Content based Music Recommendation system which recommends music to the user on the basis of similarity between songs. The songs are compared on the basis of their technical inner values like loudness, tempo and mode. The dataset used is the Million Song dataset which is the most comprehensive music dataset available. It contains metadata of one million songs.

## 6.2 Social/Economic Benefits

1. We propose a music recommendation system that recommends similar music to the user based on the songs they like.
2. Our recommender system will reduce the user effort in finding new music.
3. Our recommender system will be different from traditional recommender systems since it recommends music by analysing the technical inner values the song such as loudness, tempo, pitch etc.

## 6.3 Reflections

1. Understood the workings of Recommendation Systems in detail.
2. Understood the various data analysis techniques used to gain meaningful inference from data.
3. Became proficient in working as an enthusiastic team.

## 6.4 Future Work

1. Code optimization to improve real time recommendation.
2. Investigate using other data analysis techniques to improve accuracy of recommendation.

# PROJECT METRICS

## 7.1 Challenges Faced

1. *Dataset not normalized:* The Million Song dataset used for this project was not in normalized form. So we had to normalize the dataset.
2. *Missing values in dataset:* The dataset had missing values for certain attributes.
3. *Communication between team members:* As our team consisted of members from different groups of Computer Science department, we faced a communication gap in the beginning of the project.

## 7.2 Relevant Subjects

Table 4: Relevant Subjects

| Subject code | Subject name | Description |
|---|---|---|
| UCS 633 | Data Analytics | Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. |
| UCS 503 | Software Engineering | Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product. |

## 7.3 Peer Assessment Matrix

Table 5: Peer Assessment Matrix

| Evaluation of | | Abhimanyu | Amandeep | Harnoor |
|---|---|---|---|---|
| **Evaluation by** | **Abhimanyu** | - | 5 | 5 |
| | **Amandeep** | 5 | - | 5 |
| | **Harnoor** | 5 | 5 | - |

## 7.4 Role Playing and Work Schedule

1. Abhimanyu Sharma: Dataset cleaning and normalization.

2. Amandeep Singh: Recommendation algorithm implementation.

3. Harnoor Singh Bedi: Front end development.

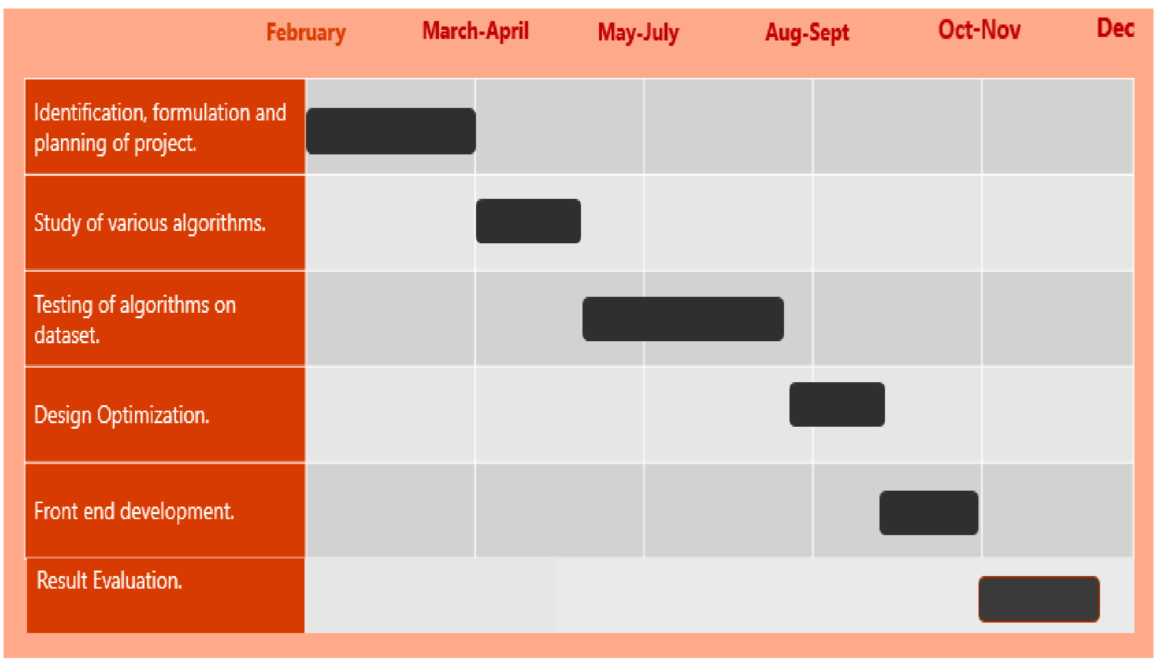



Figure 9: Work Schedule

## 7.5 Student Outcomes Description and Performance Indicators (A-K Mapping)

Table 6: A-K Mapping

| SO | Description | Outcome |
|---|---|---|
| A1 | Applying mathematical concepts to obtain analytical and numerical solutions. | Learnt about data analysis concepts. |
| A2 | Applying basic principles of science towards solving engineering problems. | Applied scientific principle of repeated experimentation to arrive at desired recommendation algorithm. |
| A3 | Applying engineering techniques for solving computing problems. | Used modern engineering tools to solve the problem of music recommendation. |
| B1 | Identify the constraints, assumptions and models for the problems. | Were able to work within the constraints of our system to provide the solution. |
| B2 | Use appropriate methods, tools and techniques for data collection. | Million song dataset from Kaggle was used for this project. |
| B3 | Analyze and interpret results with respect to assumptions, constraints and theory. | Results of the recommendation system were analyzed. |
| C1 | Design software system to address desired needs in different problem domains. | Developed a web based interface to make it easy for the user to interact with the system. |
| C2 | Can understand scope and constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. | Understood the constraints of our systems and worked within them to provide required results. |
| D1 | Fulfill assigned responsibility in multidisciplinary teams. | Each team member fulfilled his role. |
| D2 | Can play different roles as a team player. | Each team member showed a willingness to assume leadership roles and responsibilities. |
| E1 | Identify engineering problems. | Identified the problem that users face when they want to listen to new music. |

| | | |
|---|---|---|
| E2 | Develop appropriate models to formulate solutions. | Used content based model to provide recommendations. |
| E3 | Use analytical and computational methods to obtain solutions. | Used data analysis techniques to arrive at our solution. |
| F1 | Showcase professional responsibility while interacting with peers and professional communities. | Interacted with the mentors and the panel members in a professional manner. |
| F2 | Able to evaluate the ethical dimensions of a problem. | Understood professional and ethical responsibility |
| G1 | Produce a variety of documents such as laboratory or project reports using appropriate formats. | Reports were submitted on time with required format. |
| G2 | Deliver well-organized and effective oral presentation. | Presentations were delivered effectively to the panel members. |
| H1 | Aware of environmental and societal impact of engineering solutions. | Our system saves user time in recommendation of new music. |
| H2 | Examine economic tradeoffs in computing systems. | Implemented those algorithms which could run in a reasonable time on our system. |
| I1 | Able to explore and utilize resources to enhance self-learning. | Used the internet to learn various concepts of our project. |
| I2 | Recognize the importance of life-long learning. | Gained a recognition of the need for, and an ability to engage in lifelong learning |
| J1 | Comprehend the importance of contemporary issues. | Understood the various current issues in the field of music recommendation. |
| K1 | Write code in different programming languages. | Wrote code in Python programming language. |
| K2 | Apply different data structures and algorithmic techniques. | Applied data analysis algorithms to the dataset. |
| K3 | Use software tools necessary for computer engineering domain | Were able to use the techniques, and modern software engineering tools. |

## 7.6 Brief Analytical Assessment

**Q.1 What sources of information did your team explore to arrive at the list of possible Project Problems?**

**Ans:** Our team took suggestions from our mentors and referred some technical blogs and YouTube channels to arrive at the list of possible project problems.

**Q.2 What analytical, computational and/or experimental methods did your project team use to obtain solutions to the problems in the project?**

**Ans:** Our team implemented data analysis methods on the dataset in order to obtain solutions to the problems in the project.

**Q.3 How did your team shares responsibility and communicate the information of schedule with others in team to coordinate design and manufacturing dependencies?**

**Ans:** Our team shared responsibilities based on the strength of each individual. Some members had better development skills while others had better presentation skills. So the responsibilities were shared in accordance with the skill of the team members. Communication was done through a WhatsApp group.

**Q.4 What resources did you use to learn new materials not taught in class for the course of the project?**

**Ans:** The internet was our primary resource to learn new material. Whenever we encountered something we had not seen before, we searched for it on the internet and read about it on some technical blogs.

**Q.5 Does the project make you appreciate the need to solve problems in real life using engineering and could the project development make you proficient with software development tools and environments?**

**Ans:** Yes, the project does make us appreciate the need to solve real life problems. During the course of this project, we have become proficient in various software development tools and the skills learnt by doing this project will aid us in our career.

# REFERENCES


[1] B. McFee, T. BertinMahieux, D.P. Ellis, G.R. Lanckriet. “The million-song dataset challenge”. The 21st international conference companion on World Wide Web, pp. 909916, April 2012.

[2] C. Aggarwal, J. L. Wolf, K.L. Wu, P. S. Yu, “Horting hatches an egg: A new graph theoretic approach to collaborative filtering.” KDD '99 Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 201-212, 1999.

[3] D. Liang, M. Zhan, D. Ellis, “Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks”. Proceedings of the 6th ACM conference on Recommender systems, pp. 104-112, 2011.

[4] F. Aiolli. “A preliminary study on a recommender system for the million songs dataset challenge”. ECAI Workshop on Preference Learning: Problems and Application in AI, 2005.

[5] F. Aiolli, “Efficient top-n recommendations for large scale binary rated datasets”. Proceedings of the 7th ACM conference on Recommender systems, pp. 273-280, 2013.

[6] G. Adomavicius, A. Tuzhilin, “Towards the Next Generation of Recommedor Systems: A Survey of the State of the Art and Possible Extensions”. IEEE Transactions on Knowledge and Data Engineering, vol. 17(6), pp. 734-749, June 2005.

[7] G. Anthony, H. Greg, M. Tshilidzi, “Probablistic models for unified collaborative and content based recommendation in Sparse-data environments”. Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence pp. 437-444, 2011.

[8] H. Karp. “Apple iTunes sees big drop in music sales” The Wall Street Journal. Internet:https://www.wsj.com/articles/itunes-music-sales-down-more-than-13- this-year-1414166672, Oct. 24, 2014 [May 25, 2018].

[9] H. Karp. “Scores of Music Services Stream into Crowded Field” The Wall Street Journal.Internet:https://www.wsj.com/articles/scores-of-music-services-into-crowded-field-231578667, Dec. 24, 2013 [May 25, 2018].

[10] J.A. Hartigan, M.A. Wong, “Algorithm AS 136: A K-means Clustering Algorithm”. Journal of the Royal Statistical Society. Series C(Applied Statistics), Vol 28(1), pp. 100-108, 1979.

[11] M. Balabanovic, “Content-based, Collaborative recommendation” Digital Catapult, 1997.

[12] M.Sutherland. Spotify improves discovery functions”. Internet: http://www.musicweek.com/publishing/read/spotify-improves-discovery-function /072645, June 14, 2014 [May 27, 2018].

[13] M.Sutherland. “Spotify press: Information about Spotify”. Internet: http://www.musicweek.com/publishing/read/spotify-press-information-about-spotify/182375 , April 06, 2014 [May 27, 2018].

[14] M.A. Ghazanfar, A.P. Bennett, “A Scalable, Accurate Hybrid Recommender System”. Third International Conference on Knowledge Discovery and Data Mining, 2010.

[15] M.D. Ekstrand, J.T Riedl, J.A. Konstanz. “Collaborative filtering recommender systems”. Foundations and Trends in Human-Computer Interaction, vol. 4, no. 2, pp. 81-173, 2011.

[16] M.J Pazzani, D.Billsus, “Content- Based Recommendation Systems”, *The Adaptive Web*, 2nd ed. Springer-Verlag, pp. 331-370, 2007.

[17] M.J. Azur, E.A. Stuart, P.J. Leaf, “Multiple Imputation by Chained Equations: What is it and how does it work?”. Int J Methods Psychiatr Res., vol 20(1), pp. 40–49, March 1 2011.

[18] R. Burke, “Hybrid Recommendor Systems: Survey and Experiments”. User Modeling and User-Adapted Interaction, vol. 12(4), pp. 331-370, November 2002.

[19] R. Burke, “Hybrid web recommender systems”. *The Adaptive Web*, 3rd ed. Springer-Verlag, pp. 391-407, 2007.

[20] R.V. Meteren, M.V. Someren. “Using Content-Based Filtering for Recommendation”. NetlinQ Group, Amsterdam, 2000.

[21] S. Patro, K. Sahu, “Normalisation: A Preprocessing stage”, ArXiv, 2015.

[22] S.M. McNee, J.Riedl, J.A. Konstan, “Being accurate is not enough: how accuracy metrics have hurt recommender systems”. Extended abstracts on Human factors in Computing Systems, pp. 1097-1101, 2006.

[23] T. Kanungo, D.M. Mount, N.S. Netanyahu, C. Piatko, R. Silverman, “An Efficient k-Means Clustering Algorithm: Analysis and Implementation” . IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24(7), July 2002.

[24] Y. Ding, C. Liu. “Exploring drawbacks in music recommender systems–the Spotify case”.Internet:https://www.divaportal.org/smash/get/diva2:896794/FULLTEXT01.pdf, 2015 [May 26, 2018].

[25] Y. Koren. "Recommender system utilizing collaborative filtering combining explicit and implicit feedback with both neighborhood and latent factor models." AT&T Corp, AT&T Intellectual Property II LP, July 30, 2008.