****

**Executive Summary**

 Recommendation system is a program/system that tries to make a prediction based on users’ past behavior and preferences. Recommendation systems are typically seen in applications such as music listening, watching movies and e-commerce applications where users’ behavior can be modeled based on the history of purchases or consumption. We see them all around and it benefits a user in many ways because of its nature of prediction, value-add, and ease of consumption. A user would not have to spend hours to decide about a particular movie or music or a product. The relevant and useful content gets delivered to the user at appropriate times. Our aim is to build such a recommendation system using the existing tools and technologies.

**Rationale Statement**

We see an explosion of Music streaming apps these days and sometimes wonder how we get the relevant set of songs when we open the application. These are recommendation systems which are based on previous watching/listening history and feedback. Most of them work based on Collaborative and Content-based filtering. Our target is to build a music recommender app who can predict songs or podcast as per user choice, by using machine learning and deep learning algorithms. Music recommender system would be trained and tested accurately to meet the required necessity.

**Problem statement**

The goal is to implement the content-based filtering to build a comprehensive recommender system. Recommender system should suggest music based on personal preference and experience.

**List of Tasks**

1. Literature Survey
2. Dataset collection
3. Data cleaning and pre-processing
4. Data visualization
5. Algorithm selection
6. Designing Recommender Architecture
7. Implementation of the algorithms
8. Quantitative analysis of the results

**Approach:**

we will use content filtering and collaborative filtering for recommendation of music.

**Homepage Recommendations**

Homepage recommendations are personalized to a user based on their known interests. Every user sees different recommendations.

**Related Item Recommendations**

Related music recommendation are recommendations like a particular song. Users looking for a song may also see a list of related music.

**Dataset:**

Data set selected from Kaggle competition “WSDM-KKBox Music Recommendation Challenge”.

## **train.csv**

* msno: user id
* song\_id: song id
* source\_system\_tab: the name of the tab where the event was triggered. System tabs are used to categorize KKBOX mobile apps functions. For example, tab my library contains functions to manipulate the local storage, and tab search contains functions relating to search.
* source\_screen\_name: name of the layout a user sees.
* source\_type: an entry point a user first plays music on mobile apps. An entry point could be album, online-playlist.
* target: this is the target variable. target=1 means there are recurring listening event(s) triggered within a month after the user’s very first observable listening event, target=0 otherwise.

## **songs.csv**

The songs. Note that data is in unicode.

* song\_id
* song\_length: in ms
* genre\_ids: genre category. Some songs have multiple genres and they are separated by |
* artist\_name
* composer
* lyricist
* language

## **members.csv**

user information.

* msno
* city
* bd: age. Note: this column has outlier values, please use your judgement.
* gender
* registered\_via: registration method
* registration\_init\_time: format %Y%m%d
* expiration\_date: format %Y%m%d

## **song\_extra\_info.csv**

* song\_id
* song name - the name of the song.
* isrc - [International Standard Recording Code](https://en.wikipedia.org/wiki/International_Standard_Recording_Code),

### Machine Learning Algorithms:

* KNN
* LightGBM
* Deep Learning (optional)

### Tools/Technologies/Frameworks:

* Python 3.7
* Java 1.8
* Jupyter Notebook 6.0.0
* Azure ML studio
* Javascript Frameworks for UI
* NOSQL DB for Application Backend.
* Rest service in python
* Github

### Assumptions & Constraints

* It is challenging to understand the mathematical formulae, derivations and concepts behind content based and collaborative filtering.
* The understanding of the dataset and the relevant features are time consuming.

### Solution Design

* Music App: UI consist of User profile, User Preference, User Playlist. These details persist in NoSQL DB.
* Music ML: Trained Machine Learning model deploys on Azure ML.
* Music App calls rest service exposed by Music ML model for music recommendation.