1. **Introduction**

On system level point of view, the recommendation problem can be described as follows: Let U be the set of all users and let I be the set of all possible items that inside a system, such as books, movies, or restaurants. In real world applications, both item space I and user space U could be very large that contains hundreds of thousand or even millions of elements. Let f be a utility function that measures the usefulness of item i to user u as defined in below.

In this equation, s could be a ranking or score of the item i to user u. Then, for each user c belongs to C, we want to find such item s with in S that maximizes this score. More formally:

Each element u in the user space U can be described by a profile which could include various user characteristics, such as age, gender, income, marital status, etc. In the simplest case, the profile can contain only a unique ID. Similarly, each element i of the item space I is defined with another set of characteristics. For example, in a movie recommendation application, where I is a collection of movies, each movie can be represented not only by its ID, but also by its title, genre, director, year of release, leading actors, etc.

Recommender systems are designed to utilize all of these profiles in both user space U and item space I to estimate the missing score value in user-item matrix. Based on these estimations, a score based final recommendation of items to different users could be made.

In recent years, the commonly used approaches in solving the recommendation problem have been studied extensively and are usually classified into the following categories, based on how recommendations are made:

* *Popularity-based recommendations*: The user will be recommended items based on its global popularity in system
* *Content-based recommendations*: The user will be recommended items similar to the ones the user preferred in the past
* *Collaborative recommendations*: The user will be recommended items that people with similar tastes and preferences liked in the past
* *Hybrid approaches*: These methods combine collaborative and content-based methods

In our project, due to the design necessary and limitation of training data, the recommendation techniques we used are (1) popularity-based recommendations, (2) Content-based recommendations and (3) Collaborative recommendations.

1. **Technical Details**
   1. **Popularity-based recommendations**

In popularity-based recommendation methods, the score of item i for user u is estimated based on the global popularity of item i. For example, in our application, top recommend songs to user u under popularity-based method are songs which have been played and highly rated by a large number of users. The way of defining the “popularity” could be different among systems. But the usual way is to measure the rated time, average rating score and etc.

With the help of popularity-based recommendations, several critical problems in recommendation systems could be solved. For example, the recommendation to new user which also referred as cold starter is usually the popularity-based recommendations.

* 1. **Content-based recommendations**

In content-based recommendation methods, the score of item i for user u is estimated based on the historical score assigned by user u to items that are “similar” to item i. For example, in our application, in order to recommend songs to user u, the content-based recommender system tries to understand the commonalities among songs that user u already rated highly which may include style, singer and etc. Then, only songs that have a high degree of similarity to those songs are would be recommended.

More formally, let *Content-Based-Profile*(u) be the profile of user u containing tastes and preferences of this user. These profiles are obtained by analyzing the content of the items previously seen and rated by the. For the detailed implementation of this function, there are several choices in the current recommender system or information retrieval research. For example, a vector of weights W could be used as the profile by denoting the importance of each feature to user.

After summarizing the profile of each user, the final scoring function could be rewrite as:

The detailed scoring technique of item i to user u has several different choices.

* Pearson distance

The simplest way of computing distance between different items are using pearson distance which is defined as below:

* Cosine similarity

Suppose *x* and *y* are two dictionaries which contain d variables. The Cosine similarity of x and y is defined as below:

* Jaccard distance

Suppose  and  are the sets of keys from the two input dictionaries. The Jaccard distance of these two sets are defined as follows:

* 1. **Hybrid recommendations**

Although, the content-based recommendation system offer us a better understanding of user behaviors and recommend item based on their preference, there are still some problem that can’t be solved by simply using content-based recommendation system.

* Cold Starter

Inside the recommendation system, a cold starter means a new user without historical preference or a new item without many ratings from other users. For a standalone content-based recommendation system, it failed to give a reasonable recommendation since there are too less information. To deal with this situation, a hybrid recommendation system which combines content-based system with a popular based system is a common solution.

* Looping Problem

Another common problem in a single content-based recommendation system is the looping problem. The definition of looping problem is that single model recommendation system will always recommend the similar item to what user rated before. To optimize the performance of recommendation system, it’s better to do a hybrid system which could add some little changes to the recommendation result.

In our design, we believe that different user has different preference of different recommendation model. Therefore, to optimize the performance, a hybrid recommendation system which could learn the weight of recommendations from each model will be much more efficient. The learning process should be based on the user reaction to recommendation items from each model.