Milestone 1.

Music Recommendation System

1. Problem Definition

The context - Why is this problem important to solve?

Today we have several audio streaming platforms available for people. One of the most widely used one is Spotify. In 2021, it was the most popular music platform, taking a total of 31% of the market. It has over 70 million of songs. However, it is sometimes tedious for an individual to choose the relevant ones. So, there may arise a problem of choosing and listening to songs of one's interest. The problem has been handling by the platform where it will recommend the best songs to every customer based on a huge preference database gathered over time. The key is to be able to predict user's preferences so that it will benefit the customer and business itself. This can be implemented by constructing recommender systems. Recommender systems help to filter millions of songs and make personalized recommendations to users.

The objectives - What is the intended goal?

Our main goal in solving the presented problem is to create an efficient way to manage the content that consists of millions of songs and help the customers by providing quality recommendations. In this project we aim to build a personalized recommendation system to propose the top 10 songs for a user based on the history of that customer.

The key questions - What are the key questions that need to be answered?

It is important to note that it is not so easy to build recommendation systems. Recommending products, songs, movies can be difficult to interpret sometimes. People may not like the recommendations build by the platform, since they may have different moods at different periods of time.

Another question that may arise is the amount of data that we need to construct a recommendation system. It is also difficult to react with changing data and user preferences. These are the key questions and insights that need to be addressed while solving the problem.

The problem formulation - What is it that we are trying to solve using data science?

Based on the general form of the problem that we have we can now construct our main question or goal. Given a song the user is listening to, we try to predict what song will that user likely to listen to. As a data scientist we consider the Spotify dataset and by considering different relevant algorithms we will be able to solve that problem. The nature of the problem is a subclass of machine learning – recommendation engines.

2. Data Exploration

Data Description - What is the background of this data? What does it contain?

The core dataset is the Taste Profile Subset released by The Echo Nest as part of the Million Song Dataset. There are two files in this dataset. One contains the details about the song id, titles, release, artist name, and the year of release. The second file contains the user id, song id, and the play count of users. Both datasets are large: count_data has 2000000 records and song_data has 1000000 records. Below is the detailed description of the variables.

song_id: A unique id given to every song;

title: Title of the song;

release: Name of the released album;

artist_name: Name of the artist;

year: Year of release;

user_id: A unique id given to the user;

play_count: Number of times the song was played.

Observations & Insights - What are some key patterns in the data? What does it mean for the problem formulation? Are there any data treatments or pre-processing required?

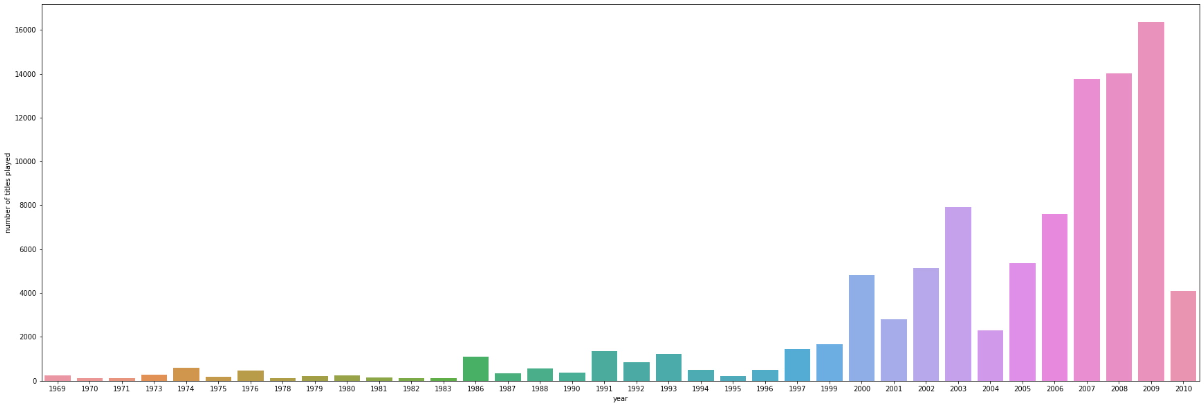
We observed that the data contains users who have listened to very few songs. Therefore, it is important to filter the data so that we have users who have listened to a good count of songs. We will be taking users who have listened to at least 90 songs, and the songs that were listened by at least 120 users. We also noted that play_count more than 5 will not add any significant value to our recommendation system, hence we may drop those records.

There are 563 unique songs, 232 unique artists and 3155 unique users in our final version of the dataset. We also analyzed the most interacted songs and the most interacted users. The song with song_id 8582 has been interacted by most users which is 751 times.

The user with user_id 61472 has interacted with the most number of songs i.e. 243 times. It is far from the actual number of songs present in the data. Hence we can build a recommendation system to recommend songs to users which they have not interacted with.

We analyzed the songs played by year as well. We can see that the most of the songs were played in 2009 which is 16351 time.

From the graph below we may observe that there are more play counts in later years from 2003 to 2010. It is clear with the rise in technology in recent years music streaming apps have been among the most in demand apps.



3. Proposed Approach

Potential techniques - What different techniques should be explored?

Once we are finished with exploring the data, we should be able to apply different models and techniques to solve our problem. First approach could be *rank based recommendation* system where we present top 10 songs based on the highest average number of play counts. This technique would be the easiest and best approach if we do not have enough data for our user.

There are other more advanced techniques which rely on the history of that user.

Collaborative Filtering recommendation system relies on user-item interaction data. It works by collecting user feedback in the form of ratings. So, in this case, it is number of play counts. For instance, we may consider user-user similarity based and item-item similarity-based techniques to recommend the songs to the user. These methods are considered to be good, since we do not need any information about the users or songs. One can implement *Model-based Collaborative Filtering*, which is a personalized recommendation system, the recommendations are based on the past behavior of the user, and it is not dependent on any additional information. This method performs well if we do not miss values in the user-item data. The last approach is *content based recommendation system* which requires information about the users and songs as title, release name and artist name.

Overall solution design - What is the potential solution design?

Potential solution design is to implement the above-mentioned algorithms on our data and try to investigate different properties of the methods. Each of them will provide slightly different outputs, our goal is to find out which of them will suit our problem solution.

Measures of success - What are the key measures of success to compare potential techniques?

The key measures of success that will be used to compare the potential models are metrics like Precision and Recall. These metrics will help us to identify the fraction of relevant recommended songs out of the recommendation list. For instance, precision is the fraction of recommended song that are relevant in top 10 predictions; recall is the fraction of relevant songs that are recommended to the user in top 10 predictions.