

Song Recommendation Challenge

Goal

Company XYZ is a very early stage startup. They allow people to stream music from their mobile for free. Right now, they still only have songs from the Beatles in their music collection, but they are planning to expand soon.

They still have all their data in json files and they are interested in getting some basic info about their users as well as building a very preliminary song recommendation model in order to increase user engagement.

Challenge Description

You are the fifth employee at company XYZ. The good news is that if the company becomes big, you will become very rich with the stocks. The bad news is that at such an early stage the data is usually very messy. All their data is stored in json format.

The company CEO asked you for very specific questions:

- What are the top 3 and the bottom 3 states in terms number of users?
 - What are the top 3 and the bottom 3 states in terms of user engagement? You can choose how to mathematically define user engagement. What the CEO cares about here is in which states users are using the product a lot/very little.
 - The CEO wants to send a gift to the first user who signed-up for each state. That is, the first user who signed-up from California, from Oregon, etc. Can you give him a list of those users?
 - Build a function that takes as an input any of the songs in the data and returns the most likely song to be listened next.
That is, if, for instance, a user is currently listening to “Eight Days A Week“, which song has the highest probability of being played right after it by the same user? This is going to be V1 of a song recommendation model.
 - How would you set up a test to check whether your model works well?
-

Data

data - all the data is here. Each row represents a song that was listened by a user.

Fields:

- **id** : it is unique.
- **user_id** : user id who listened to a given song.
- **user_state** : where the user is based.
- **user_sign_up_date** : when the user signed-up.
- **song_played** : the song that was listened.
- **time_played** : at which time the user started listening to the song (local time).