# Computational Thinking

# Group Project:

# Spotify

Vrije Universiteit Amsterdam

**Project Title: Spotify**

The following problem is modelled after real-life algorithms.

Assume you want to create a program that mimics Spotify's "Discover Weekly." Your program will of course be much simpler due to time and resource constraints. Discover Weekly generates suggestions for users of Spotify: a few new songs every few days, possibly songs that the users have not heard but will most probably enjoy.

We will assume the following in this problem:

You have 100 users, who use your main app, which resembles Spotify. You are launching a new feature for an app imitating "Discover weekly". This new feature will also suggest to the users what songs they can listen to, that they may like. This means that you will be looking at what songs people played in the app and we will suggest new songs to them, in the new feature, based on what we know. The suggested 5 songs may be songs the user have played.

**Step 1: (week 1 after the day of the launch) Individual choice**
Each of your 100 users is trying the new Discover feature in your app for the first time, in the first week. The feature will generate songs that he or she may like. You need to generate 5 songs. Assume this is the first week in which you launched your app on google store.

Assume that all users start using the feature on day 1 and no one downloads it later.
The 5 songs that will be generated for the first time we pick in the following way:

You already have 100 pre-made playlists, with 50 songs in each. Your task is the following: if you find a playlist which contains 3 songs that the user has listened to already and 3 he has not heard yet, you select this playlist. From this playlist, your algorithm picks any 5 songs.

**Step 2: (week 2) Genres**
In the second week you will use a different approach:
A user opens the app and is looking for new suggestions on "Discover."

There are three types of music only: pop, rock, and techno.

You want to find which style the user has listened to the most, of those three. Define in your own terms what this should mean. For example, you can say that if the user has listened to 20 rock songs and 2 techno songs and 3 pop songs, this is a rock music fan. Or you can work with percentages: you can find a user that played 80 % techno and 20 % pop. You can make your algorithm disregard a difference of 10 % and say that if a user played pop 45 percent of the time and rock 50 percent of the time, the difference is small and both styles should be considered equally important to him or her. Outline your scale and assumptions in your work.

In the second week, your algorithm should suggest 5 songs again, this time based on the above-mentioned, i.e. based on style.

It is possible that the program suggests one song in one week and again the same song in a following week. A more refined algorithm, not required here, could account for repetition.

**Step 3: (week 3) Shifts**
In the third week after the launch, you want to account for a user's mood shift. We have a few types of songs: "happy", "party", "calming" and "lounge." Define in your own algorithm if you think one song can be classified as two of those at the same time.

This week you need to select 5 more songs to suggest to each user, again. If last week the user was playing a lot of songs from one of those types, this week we want to suggest more of the same type. Define, in your own terms what this should mean: if the user listened to 3 "calming" songs, should we provide 3 more without considering other factors? What if he or she also listened to 3 "party" songs and 4 "lounge" style songs?

Elaborate on a successful approach in this step. Consider while-loops, for example.

**Literature:**

https://blogs.cornell.edu/info4220/2016/03/18/spotify-recommendation-matching-algorithm/

**Additional assumptions and suggestions:**
You can assume that while you are creating your program, your computer can generate random numbers with value 0 to 1 for you. Thus, if your computer generates value 0.5, you can use it, by multiplying it by ten to get a value of 5.

Suppose you want to simulate a program that provides a simple alternative to Spotify and their feature "Discover Weekly," following the example steps above.

You are allowed to approach this problem creatively, making your own assumptions of what is allowed. For example, can we combine the approaches in step 1, step 2 and step 3 somehow in later weeks? Or after week 3, do we want to make suggestions only based on the approach in week 3 ignoring what we did in week 1 and 2?

## Your task:

Familiarize yourself with the idea of using an **algorithm, pseudocode, and Python** before you start with this project.

1. Create an algorithm that allows the simulation to exist. (35 pts)
2. Translate your algorithm to a flowchart. (30 pts)
3. Implement your algorithm in Python. (35)