
Bilkent University

Department of Computer Engineering

## CS 464 Introduction to Machine Learning

Building Spotify Playlists Using Machine Learning

# Progress Report

**Group 9**

Arda Akça Büyük

Can Aybalık

İsmail Yavuzselim Taşçı

Oğuz Kaan İmamoğlu

Ömer Altuğ Sevimay

**Instructor:** Ercüment Çiçek

**Teaching Assistant:** Furkan Özden

# 1. Introduction

We are trying to create a recommended playlists with songs that have similar features to a single song or multiple songs, or even an artist chosen by the user. Having said that, we are aimed to create new playlists that satisfy user tastes. While we are trying to achieve this, we will utilize machine learning algorithms and methods. And the dataset that we will use will help us to detect similarity between songs according to their features and measures. We plan to measure how successful the results of the project will be with the surveys we will make to users.

For the project, we will use a dataset named “Spotify Dataset 1921-2020, 160k+ Tracks” which is on Kaggle[1]. This dataset includes many different features of songs that are on Spotify. Some of the features represent the mood of a song such as danceability, energy, valence, tempo. These are measures from 0.0 to 1.0 except tempo, which is the BPM (Beat Per Minute) of the song that typically ranges from 50 to 150. Some of the features represent the properties of the sound of a song such as loudness, speechiness, instrumentalness, and mode. These are also some measures from 0.0 to 1.0 except loudness (it ranges from -60 to 0). And also, there are some contextual features of a song in the dataset, which are acoustic and liveness. In the dataset, there are also some properties that are independent from the audio features of a song which are id of a song, duration of a song, year of a song, and whether it has explicit content or not.

Three machine learning algorithms which are kNN, Random Forest Regression and Linear Regression will be used in this project. Also, we will implement a website which will connect to the user’s Spotify account and with the help of our algorithms, it will create a recommended playlist in that account.

As a final result, we are to have a product similar to dubolt.com, which generates playlists according to user choices [2].

# 2. What We Done So Far

In this section, we presented what we have done so far with our project.

## 2.1. Plotting the Feature Correlations

In order to get important insights about the songs' correlations, we plotted our data with respect to some features such as song mood-danceability and energy-acousticness. In this manner, while recommending songs to the user, we will be able to get insights from the plots we have created and even increase our consistency with the help of correlations between features.

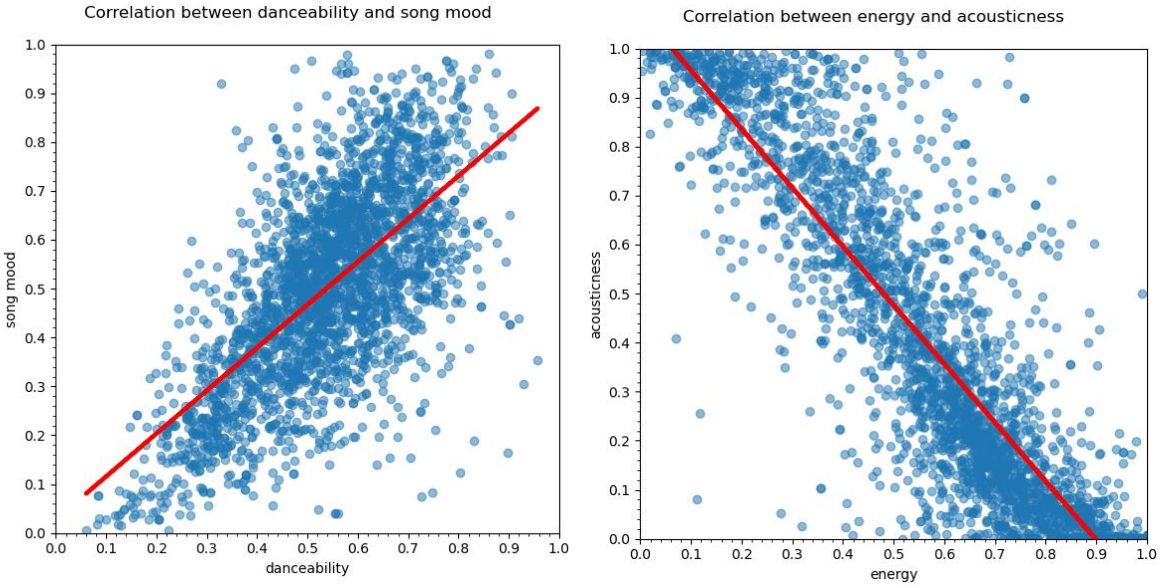



*Figure 1: Correlations between danceability-song mood and energy-acousticness*

For instance, Figure 1 shows that the positive correlation between danceability and song mode indicates that when the user listens to songs with high danceability, we can recommend songs with a high song mood. Also, not all features will have positive correlations. Features such as energy-acousticness will give negative correlation so if the user listens to songs with high energy, we can recommend songs with low acousticness. The red line is the result of the linear regression of the two features in the plot. By looking at the angle of the red line, we can understand what degree of correlation there is between features. However, some features may not show any correlation at all. For example, there is little talk of a positive or negative correlation between liveness and spechiness, so it will be difficult to make predictions between the two.

## 2.2. Getting Data from Spotify Web API

To accomplish the aim of the project, we need to access and manipulate the user's Spotify account. With this access, we will get the currently playing track's data from the user and evaluate this track with multiple machine learning algorithms. After that, we will create a playlist that contains recommended songs from the algorithm.

- **Creating a new app**

To use the Spotify API in our program, firstly we need to create a new app in "Spotify for Developers". This is essential because it assigns a necessary Client ID and Client Secret ID to the developer. With these IDs, we can get an authorization from Spotify to access Spotify API and the user's Spotify information such as albums of an artist, all songs of an artist, user's currently playing song, user's playlists or user's saved songs.

- **Some examples from Spotify API**

Here, there are some examples about what information we have accessed from the Spotify API.

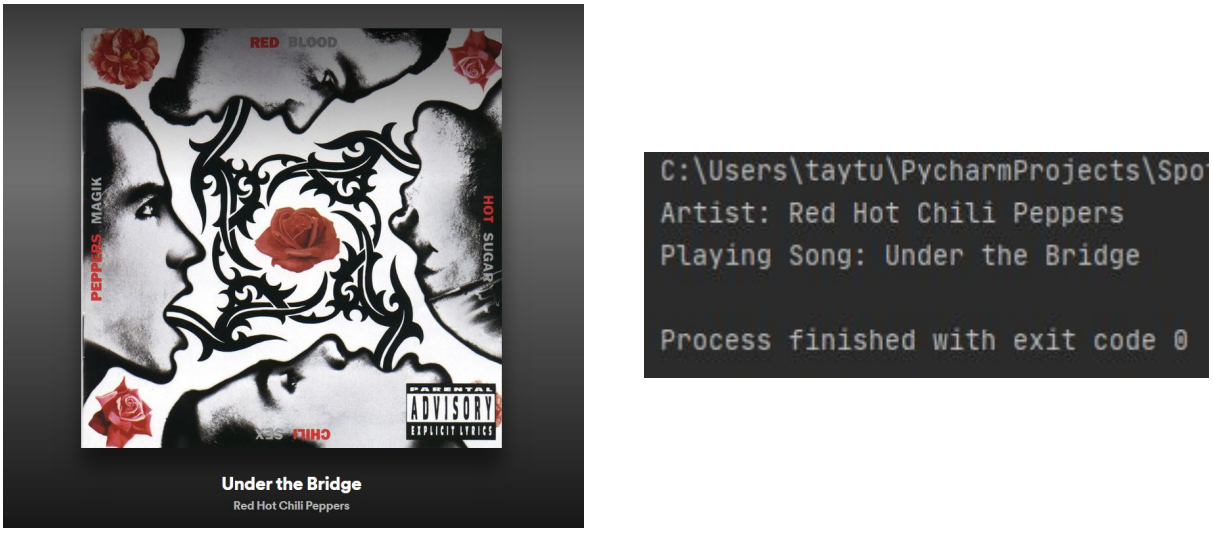



*Figure 2: Obtained data from the user's currently playing song*

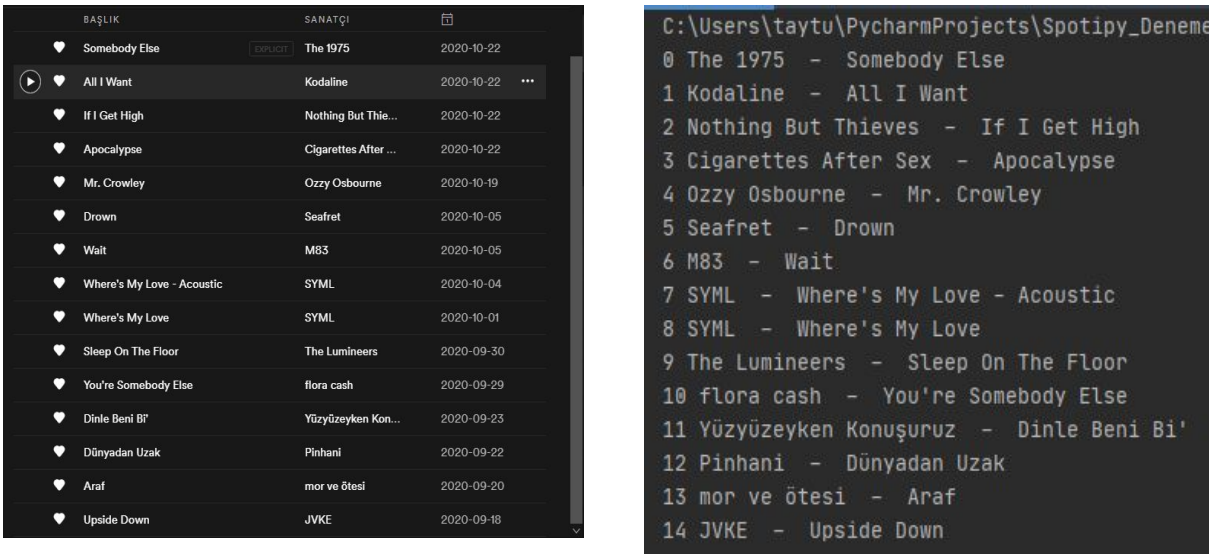



*Figure 3: Obtained data from the user's saved songs*

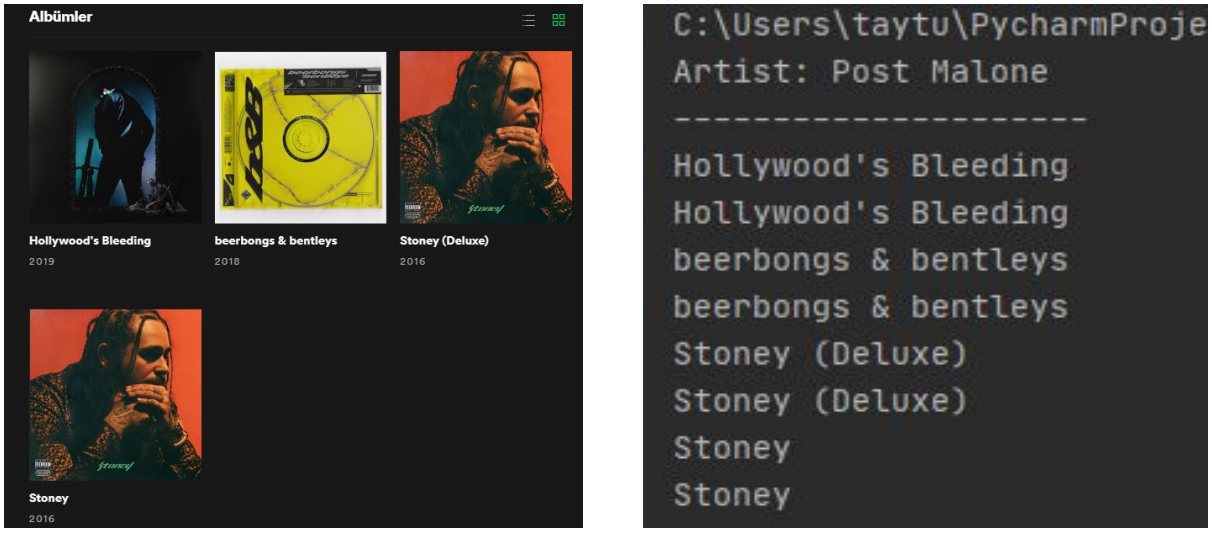



*Figure 4: The list of all albums of the artist "Post Malone"*


*Figure 5: Creating a Playlist for the user*

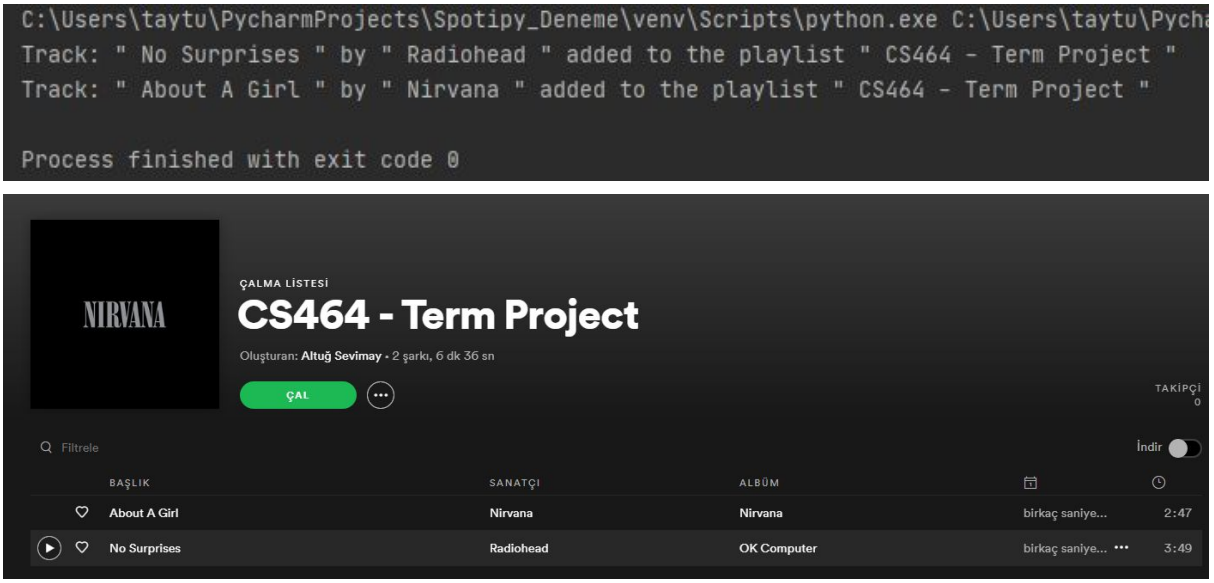



*Figure 6: Adding tracks to the user's newly created playlist*

## 2.3. Determining Machine Learning Algorithms we will use

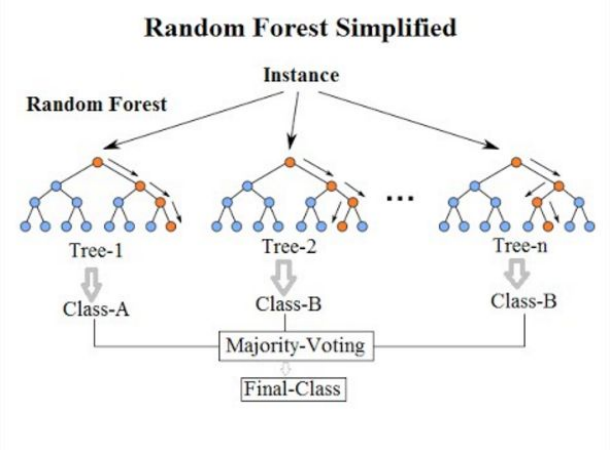
In this section, we presented the algorithms that we are going to try in our project.

### 2.3.1. kNN

K-Nearest Neighbors algorithm utilizes the feature similarity between existing and new data based on the notion of distance which is very appropriate for our recommendation system. This algorithm will allow us to find other songs that are closest to the songs the user is listening to, so we can create a highly functional playlist.

### 2.3.2. Random Forest Regression

In the Random Forest Regression model, data is divided into several sub-sections. Each subsection of data is used to generate one prediction. Then these predictions are combined to finalize the prediction. We will use this model to produce N different playlists and rank the scores of these playlists. Then by combining first n playlists will determine the final playlist recommendation.




### 2.3.3. Linear Regression

We will train the model with the data of the user, and we will predict an average of the features of the songs accordingly. Since we won't probably have a song that matches those exact features, we will pick the closest song that has similar features with the prediction.

## 3. What Remains to be Done

From now on, we need to implement algorithms that we decided on. After trying various algorithms, we will measure their success and pick the best algorithm to use in our application. Other than this effort, we will implement a presentation layer, most probably a website, to allow our users to use our algorithm to create a new Spotify playlist according to their taste.

## 4. Division of the Work

Arda Akça Büyük - Website and KNN algorithm implementation

Can Aybalık - Random Forest Regression and Linear Regression

İsmail Yavuzselim Taşçı - Website and KNN algorithm implementation

Oğuz Kaan İmamoğlu - Random Forest Regression. data manipulation with pandas

Ömer Altuğ Sevimay - Spotify API and Linear Regression

# 5. References

[1] Kaggle spotify dataset. [Online]. https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks?select=data.csv. [Accessed 16-Nov-2020].

[2] Dubolt. [Online]. https://dubolt.com/. [Accessed 16-Nov-2020].