**Project Ideation**

Create a Spotfy playlist on a weekly bases that suits the user's music taste based on music dataset containing songs that they have been listening to during the previous week.

The playlist can be modified by the user. Which means that they can remove or add songs to it based on their interests.Starting with a song that they love the most, the playlist will contain 50 songs per week and it will end with a song that they have never listened to.

**Problmem Analysis**

**Inputs**

Since we are going to be using python as our program language, we are first going to install required packages that’ll be used to create a playlist.These are spotipy, matplotlib, pandas, scikit-learn and pydotplus.

I will then read playlists and modify them using the Spotfy Api and for this I need to get an Spotfy ID. I will use instructions from <https://developer.spotify.com/documentation/web-api/reference/#/operations/get-playlist>

**Outputs**

After gettingn the access token, we use the playlist ID from the playlist to get a list of tracks showing their names. Additionally we can extract the Audio features of the tracks in the given playlist. Using pandas, we can then create a dataframe with column names trackID, Track Name and Audio Features.

A reccommended playlist will come from the prediction made by the specified classfication algorithm below. It will be stored as a dataframe with the song features (Track Name and Artist).

**Algorithm**

In order to make a distinction of the songs that a user likes and do not, I will use Classification. I then create a dictionary with a pair key value of the song as a key and whether the song is categorized as liked(1) or disliked(0). The number of elements in the dictionary are going to be 50.

I then add an additional column taking only the values of the dictionary into a new column called "Rating" into the dataframe, which indicates whether the person likes the song or not (as ones and zeros respectively).

In order to attain the reccommendations which the output of our playlist, I am going to use, the k-nearest neighbors (KNN) algorithm using KNeighborsClassifier from sklearn. With our data seperated into 80% training set and 20% testing set we will fit our model and determine the accuracy using the confusion matrix with N=4 (N being the number of the nearest neighbors)

To run the classification algorithm I am going to use information from the documentation at <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html> and an example from <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>