The largest project that I have worked on was a personal project of mine which implemented a song recommendation system for playlists on Spotify. I used Python and Spotify's Web API throughout this project. This project was aimed at recommending new songs to a user's playlists based on the audio features of the existing songs in the playlist. The code first loads existing songs in a user's playlists and extracts data about the song's audio features using HTTP requests and sending JSON object. It then runs various tests such as hypothesis tests, p - value tests and builds LASSO, Ridge Regression, and Elastic Net Logistic Regression models to find which variables affect its probability of being included in the playlist. Finally it loads new songs from Spotify and runs the generated regression model on the new songs to accurately predict which songs to include in the user's playlist. The selected songs are then added to the playlist by again using HTTP requests.

I am going to expand on the project described above. The main motivation behind the above project was that I was not very satisfied with the songs recommended by Spotify to my gym playlist since I regularly work out and need a constant influx of new songs to keep the pump going. So, I decided to make an algorithm of my own which could possibly provide a better and more personalized selection of songs. Since the probability of a song being included in a user's playlist is affected by different factors for different users, a new regression model had to be made for each individual user. The most difficult problem I faced throughout this project was finding the most parsimonious model - finding the exact set of variables which give enough power to the model to accurately predict which sings to be added to the playlist. I had to make multiple models, add different weights to the regularized regression models, and compare the slopes of the variables in the different regularized models coupled with the hypothesis and p-value tests to finally choose the most appropriate variables. Another challenge I faced was the initial authentication process of requesting access to a user's playlist. This was done through a process called OAuth 2.0. Since I had no prior experience of dealing with such authentication processes, I first had to learn how this kind of authentication worked and how to use python to automate this. I did this by watching multiple YouTube videos on OAuth 2.0 and reading various online articles. In the end, however, I was able to making a functional model which could predict with considerable accuracy, whether a song should be added to a user's playlist or not.