# Project Proposal

Colin Legge, Jacob Ericson, Suqian Wang

## Introduction

The problem we decided to solve was a recommendation problem. We decided to provide a recommendation to users of songs or playlists that they may be interested in based on our library of songs. We will be using the data gathered from Spotify and youtube as our source for providing our recommendations. The program will work with the user searching for various keywords, artists, genres, or songs. Our program will then return a list of songs including the searched keyword. If the user decided to search for a particular song we will return related songs, their youtube videos, and playlists of similar interests. The returned results are our recommendations to the user on what that user would be interested in based on our library of data. Our recommendations will also return to the user the songs within playlist containing the searched song. Our program will solve our problem because it will provide a recommendation to our users for songs, playlists, videos, and related songs based on their searches.

## Source Data

We will be using spotify playlists' songs data

## Tasks

- We will crawl the data includes playlists IDs and names of Spotify
- We will crawl the info of songs in the playlists above
- We will create out text search engine using Solr
- We will index our data using Solr
- We will cross compare song data including artists, album, and other info
- We will analyze the difference and the similarities
- We will make out query based on our objective
- We will build a web application for our search engine and database

## Algorithms/Techniques/Models

- Libraries used: Spotipy, JSON
- Techniques: Inverted index, Phrase queries, Wild-card queries
- Algorithms: Merge algorithm (Intersecting two posting lists), Porter's algorithm, Some document correction algorithms, Soundex

- Models: Boolean Retrieval Model & Extended Boolean Models, Statistical language model, Weighted edit distance

## Evaluate Method

To evaluate the correctness of our program and it's functionality we will be conducting multiple tests on its searching capabilities and recommendation quality. We will test our search engine using some specific songs, playlists, and various keywords. We will also test the randomness of the samples songs, playlists, and keywords we select. In order to fully test the capabilities of our programs recommendations. That way we don't get results that are not related to our searches and songs.

We will check the correctness of our program based on the feedback. Get user feedback to see the quality of recommendations. We will test the quality of user reviews of our web application. From their recommendations, we can make further improvements to our search algorithms or recommendation algorithms. We will check the recommended youtube video shows the correct music video, rather videos created by other users. Our program should not be returning lyric videos or cover songs by other artists to our user.

The success of our tests will come from the programs ability to satisfy our user given our program's recommendations. We will test our search algorithms by how they can correctly match keywords to titles of songs. We will also test the functionality of our search algorithms by how many matches and search results are returned to the user. If a keyword or song is entered, we do not want only one search result is returned, when multiple results can be matched.

## Goal

By the end of the semester, we will be able to build a web application that can perform keyword search for songs. User will be able to get recommendations of songs, playlists, and trend. User will be able to listen to songs and watch their tailored video.