Final Project Report
Christy Sato
IST652 Scripting

**Data Description**

The sources of the data I analyzed was from Spotify API and Twitter API. For both Spotify and Twitter, I needed to create a developer app so that I can access both APIs. By making an app, I am given a Client ID and a client secret so that I can extract the data or tweets from the APIs. The data I collected from both APIs is about study playlists. The Spotify API was available from a package called spotipy, which provided information on what features in songs make it a good song to listen to while studying. The dataset I created from the collection of Spotify API includes information like track and playlist id, date added, track name, artist name, audio features (like danceability, energy, loudness, etc) and many more. In total, there were 2,311 observations and 28 columns. The below image shows the data types of each column. Most columns were objects and floats, and a few being integers.

```
data.dtypes
```

```
track_id             object
playlist_id          object
date_added           object
track_name           object
first_artist         object
danceability        float64
energy              float64
key                   int64
loudness            float64
mode                  int64
speechiness         float64
acousticness        float64
instrumentalness    float64
liveness            float64
valence             float64
tempo               float64
type                 object
id                   object
uri                  object
track_href           object
analysis_url         object
duration_ms           int64
time_signature        int64
lyrics              float64
neg                 float64
neu                 float64
pos                 float64
compound            float64
dtype: object
```

To further understand each column, here is a description of each column. Track_id is the id of the track, playlist_id is the id of the playlist, date_added is the date of the track being added to the playlist, track_name is the name of track, first_artist is the name of the artist, danceability is how suitable a track is for dancing (the higher the value, the more danceable), energy is a measure of intensity and activity (energetic songs are fast and noisy), key is overall key (pitch) of the song (ex: 0 = C, 1 = C♯/D♭, 2 = D), loudness is the quality of sound that is primary psychological correlate of physical strength (amplitude), mode indicates the modality (major or minor) of a song, speechiness detects the presence of spoken words, acousticness is the confidence measure of how acoustic a song is, instrumentalness is whether a song has vocals (closer to 1 the song does not have any vocal content), liveness detects the presense of audience in the recording, valence describes how positive a song is (the closer to 1 is more positive and cheerful), tempo is the beats per minute (speed or pace), type is the object type “audio_features”, id is Spotify ID for the track, uri is the Spotify URI for the track, track_href is a link to the web API, analysis_url is the URL to access full audio analysis, duration_ms is duration of song in milliseconds, time_signature specifies how many beats are in each bar (measure), lyrics is the lyrics of the song (if provided), neg, neu, and pos is how positive, negative, or neutral the lyrics (sentiment analysis) seem to be. Compound is the blend of two or more song structures.

From the Twitter API, I had to first create a new mongo database and extract tweets from the "run_twitter_simple_search_save.py" script. I was able to collect 74 tweets that had the hashtag '#studyplaylists', '#studysongs', and '#studymusic'. I then created a pandas dataframe that had the following information: the tweet id, the user's screen name, the text of the tweet, the language, the place, and the retweet and favorite count on the tweets. 4 of these columns (user, text, lang, place) were objects and 3 of them (id, retweet_count, favorite_count) were integers.

**Data Preprocessing**

Before I can start my analysis on the datasets I had to do some preprocessing. I first dropped columns I did not believe I needed for further analysis. I dropped the following columns: track_id, mode, type, id, uri, track_href, anaylsis_url, time_signature, lyrics, neg, neu, pos, and compound. Most of the columns had missing values, and was not the main focus of my anaylsis. I then changed the values of the playlist_id to the names of the playlists I collected from, so that I can see which track came from which playlist. I then renamed the playlist_id column to playlist_name. Once the main dataframe was cleaned, I then created two separate dataframes, one with all the study playlists (called studydf) and another with the other genre playlists (called myplaylists).

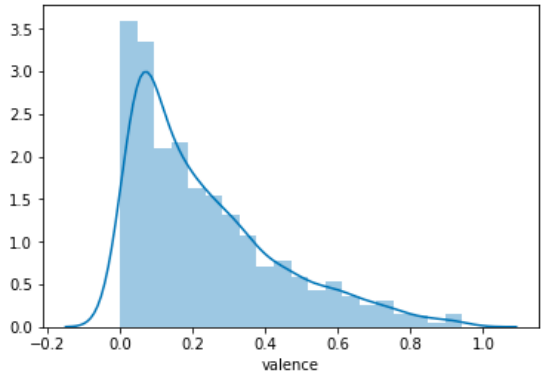
For the twitter dataframe, the preprocessing steps I did was first tokenize all the tweets. Each word in all the tweets become their own token. However, a lot of times other unnecessary symbols and characters becomes tokens as well. So to remove them we can filter out those symbols. I have also lower cased all the tokens and removed English stop words.

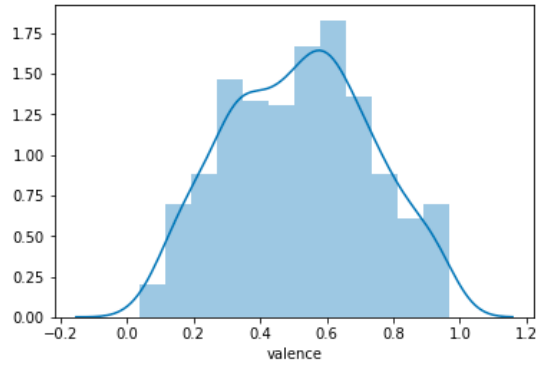
**Methods of Analysis**

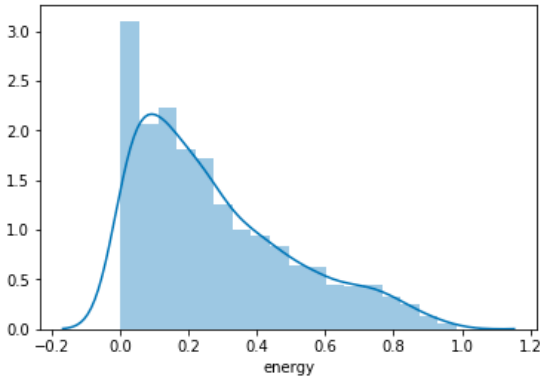
The first question I wanted to ask is what features in songs make it a good study song? To answer this question, I used the studydf that has all the study playlists. I took a look at all the audio feature columns to get a better understanding of the features of study playlists and songs. By grouping the dataframe by the playlist name and get the mean values of each audio feature for each specific playlist, I had a better understanding. As a result of the grouping, the index of the dataframe became the different playlist names, and then for each playlist, there was a mean value for the various features like danceability, energy, etc. The resulting output was 16 rows (I ended up not using all the 20 study playlists due to not get overwhelmed with more numbers for the viewer and presentation) and 11 columns. What I focused on was danceability, energy, loudness, and valence. By taking a look at the different playlist and their values, I can see certain study playlists have low danceability, energy, loudness, and valence. While other study playlists actually have rather high danceability, energy, loudness, and valence. So I concluded that there is no definitive answer that is black and white. Different playlists had different types of songs, which explains why the playlists vary. It all depends on the user's

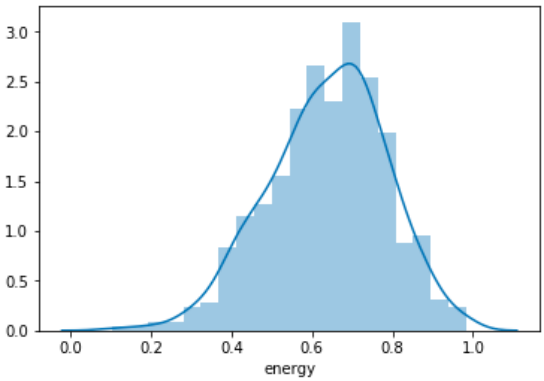
and their preference. Personally, when I am trying to be productive and listen to music, I need low energy and danceability so that I can stay focused and not burst out into song. But others may actually need high energy songs so that they do not feel tired while studying.

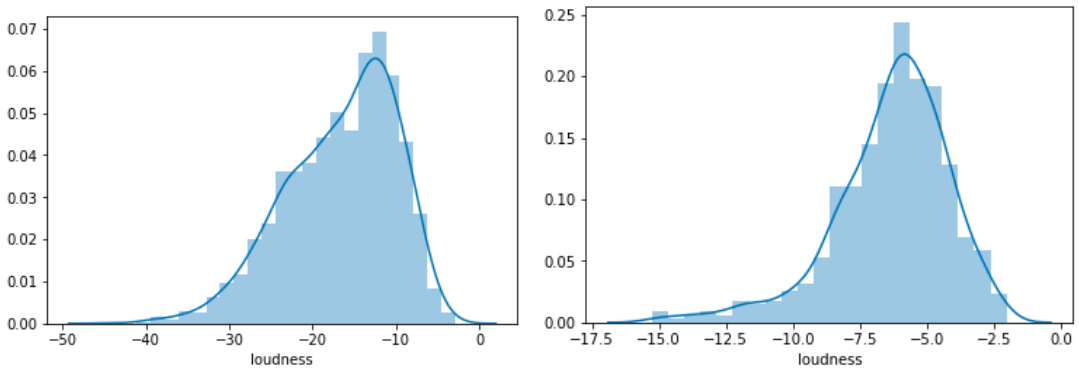
The second question I was curious about is the comparison of features in both the study playlists and other genre playlists that I usually listen to. Here I used the studydf and myplaylists to compare the two. I took a look at the two separate dataframes that both had the different playlist names and the audio features. The output I used for analysis is both the studydf as mentioned in the first analysis question and the myplaylist dataframe that had only 8 rows and 11 columns. Just by looking at the two datasets, I can visually see that myplaylists had overall higher mean values in danceability, energy, loudness, and valence. But just looking at numbers was not enough, I wanted to look at them illustrated in histograms. So by importing seaborn, I was able to create graphs and compare the two datasets.






The left graphs are for the study playlists, while the right are the other genre playlists. Looking at these, helps makes conclusions in comparing the two datasets. For the study playlists you can see that the songs had mostly low valence and energy while myplaylists had higher energy and valence songs. In conclusion, by comparing the study playlists and my playlists, I usually listen to songs that have high energy, are considered more positive, and loud, while the study playlists are less cheerful, low energy, and less loud.

Lastly, the third analysis question is what twitter users are saying about study playlists. In doing so, I wanted to take a look at the frequency of words in tweets to see what most people are saying. To do this, I used the tokens I have created from the tweetdata dataframe. The tokens are then counted, to get the most frequent words. The final output was a list of the top 30 most common words and the number of times it showed up in all the tweets.

```
https 63
music 34
rt 25
electronic 16
remix 16
song 15
tasterdaysontour 15
edm 14
studymusic 14
electro 13
songremix 12
dubstep 12
bassdrop 12
songwriter 11
new 10
now 10
study_music_cam 9
got 9
fluorescenceboi 8
f4f 8
focus 8
study 8
deep 7
nowplaying 7
day 7
lofi 7
follow4follow 6
ambient 6
ta 6
track 6
```

The most notable thing I noticed was that the words; electronic, edm, electro, dubstep, and bassdrop were part of the top 30 words. Due to this, I was able to conclude that most twitter users enjoyed listening to the genre edm, which usually has high danceability, energy, and loudness. So the study playlists that had high levels of those probably would work best for them.

**Final Project Program**

The program I created was done on jupyter notebook. With a few steps done on terminal, which is instructed in my jupyter notebook when needed. The first few steps, I attempted to get a livestream from twitter. But in the end, I was not able to get any final conclusions from it, so that can be ignored. I then worked on the Spotify API which I have never worked with, so I had to research a lot to complete this project. There are a few links I put for references I used to educate myself in this topic. Twitter API was also used in this project as you can see later in the program. But for both APIs the necessary data cleaning, exploration, and final outputs and analysis were done. I was most comfortable using pandas dataframes because visually it is easy to understand. There are some trial and error outputs I made to see if I can make any further conclusions on, but in the end were inconclusive. For example, I looked at a specific artist Khalid, and noticed the artist was seen in both a study playlist and my playlist dataframes. But no further analysis was done on that topic. I also tried to make separate dataframe that I thought was an intense studying playlist, but again, nothing was interpreted from it. But it all helped decide what I wanted to include in my final presentation.