SI 206 Fall 2019 Final Project Report

Team Name: Emotify

Helen Song, Jennifer Tsai

GitHub Repository Link https://github.com/jltsai/finalproject

Goals for Project

The major goal of this project was to create a program that would recommend songs as a mood booster based on the sentiment and emotion levels calculated from a list of strings. Our plan was to utilize two APIs (ParallelDots and Spotify) in order to acquire the emotion and sentiment scores for a string of text as float values, then recommend a song based off of those scores by setting the target values for energy and valence levels in a song. These target values would not necessarily be the actual values of energy and valence levels for the songs recommended. The program should also be able to assess any length of text. In other words, the higher the level of bored emotion and negative sentiment in the string, the greater the target energy and target valence values are set for recommending the songs.

Goals Achieved

The two APIs we used for this project are Spotify and ParallelDots. Spotify is an international media services provider, mainly for songs and podcasts. ParallelDots is an artificial intelligence sentiment and emotion analyzer.

The Sentiment API from ParallelDots is able to determine the overall sentiment of the text by assessing the confidence score of the sentiments (positive, negative, neutral). We chose to focus only on the negative sentiment of the text when making calculations from the database. The Emotion API is able to determine the overall emotion by assessing the confidence score of the emotions (bored, sad, happy, angry, excited, and fear). We chose to focus only on the bored emotion levels.

There are no limitations to the number of characters that are able to be passed to both APIs, which satisfies our goal of being able to assess any length of text. The confidence scores outputted from ParallelDots will determine the parameters (target energy and target valence) for the Spotify-recommended songs. The more negative sentiment there is, the higher the target valence set for the recommended song. The more bored emotion there is, the higher the target energy level set for the recommended song. The Spotify API then is able to recommend songs based off of those target values, although the *actual* valence and energy levels of the recommended songs may differ. Ultimately, this program allows us to achieve our goal of recommending songs that would be mood boosters for those feeling negative emotion and sentiment.

Problems Faced

Most of the problems we had were resolved simply by using our online resources. Some problems faced included limited access to API keys daily, understanding syntax for SQL, and debugging issues for errors such as TypeError, NameError, and SyntaxError. Initially, we had trouble converting a cURL command syntax to Python when requesting the Spotify API. We also had an issue with limiting data to 20 items stored every time our code runs and making sure that duplicate data was not printed into the database after running our main file more than once.

File with Calculations from Database

Note: emotifyoutput.txt is in the zipped file

```
(Recommended Song, % Change SONG Actual Valence | TEXT Negative Sentiment, % Change SONG Actual Energy | TEXT Bored Emotion)

(<What Is Love> by Kiesza, 2.6315789473684235, 18.30715811243856)
```

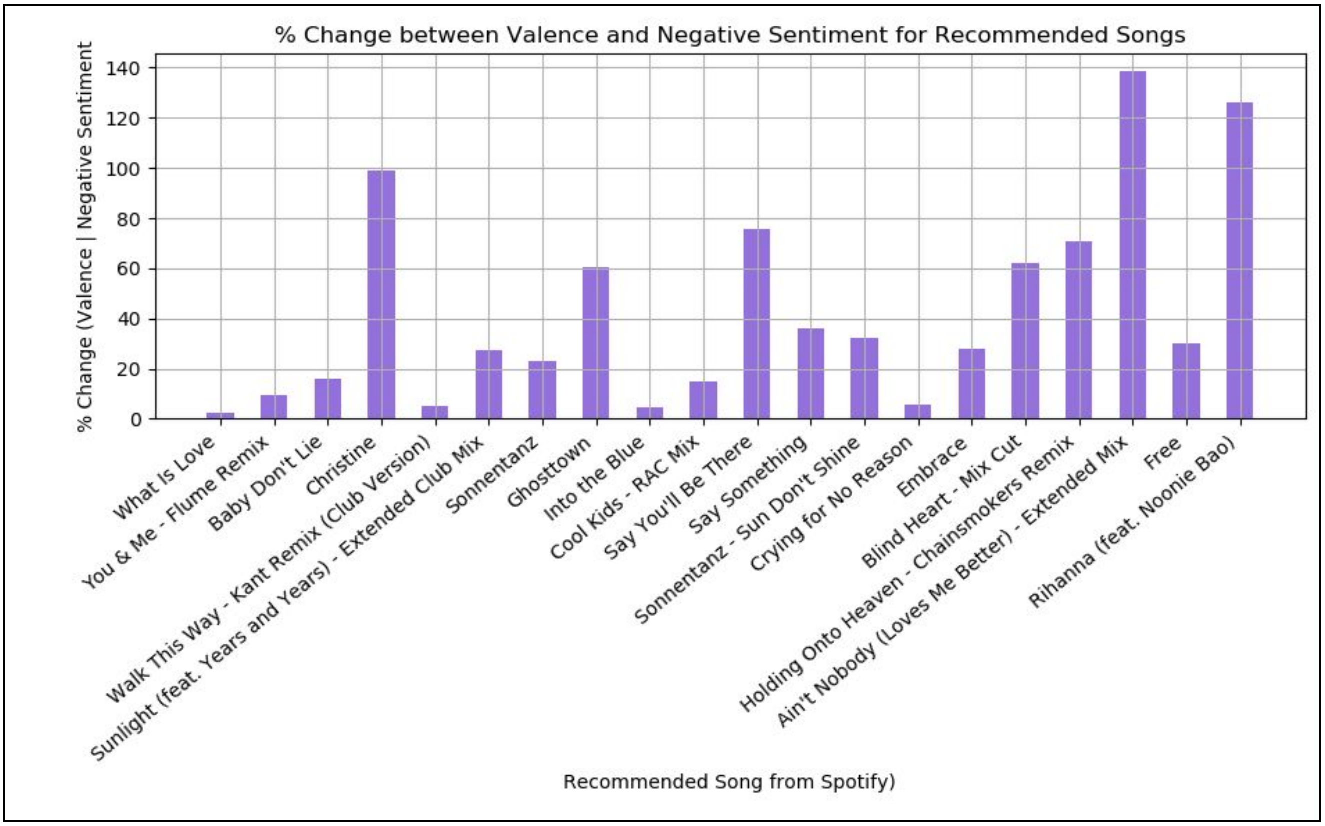
```
(<You & Me - Flume Remix> by Disclosure, 9.649122807017541, 31.20598289394007)
(<Christine> by Christine and the Queens, 99.12280701754386, 28.905555544372923)
(<Baby Don't Lie> by Gwen Stefani, 15.789473684210527, 44.26912391469638)
(<Walk This Way - Kant Remix (Club Version)> by MØ, 5.263157894736847, 47.226816221282704)
(<Sunlight (feat. Years and Years) - Extended Club Mix> by The Magician, 27.19298245614035, 43.2832264791676)
(<Sonnentanz> by Klangkarussell, 22.807017543859658, 26.85160255368797)
(<Ghosttown> by Madonna, 60.57017543859649, 44.18696579506898)
(<Into the Blue> by Kylie Minogue, 4.824561403508776, 54.70320510737594)
(<Cool Kids - RAC Mix> by Echosmith, 14.912280701754387, 54.21025638961155)
(<Say You'll Be There> by MØ, 75.43859649122808, 36.54626066972095)
(<Say Something> by Karen Harding, 35.96491228070175, 50.59529912600603)
(<Sonnentanz - Sun Don't Shine> by Klangkarussell, 32.01754385964912, 44.104807675441585)
(<Crying for No Reason> by Katy B, 5.701754385964906, 56.59284185880609)
(<Embrace> by Goldroom, 28.07017543859648, 55.77126066253211)
(<Blind Heart - Mix Cut> by CAZZETTE, 62.28070175438596, 49.527243570849855)
(<Holding Onto Heaven - Chainsmokers Remix> by Foxes, 71.05263157894736, 49.03429485308546)
(<Ain't Nobody (Loves Me Better) - Extended Mix> by Felix Jaehn, 138.5964912280702, 36.62841878934835)
(<Free> by Rudimental, 30.263157894736842, 48.13055553718409)
(<Rihanna (feat. Noonie Bao)> by Clean Bandit, 125.87719298245615, 40.07905981369907)
(<You & Me - Flume Remix> by Disclosure, 14.965986394557818, 12.69776588914199)
(<What Is Love> by Kiesza, 24.489795918367342, 7.211670877419304)
(<Christine> by Christine and the Queens, 54.42176870748301, 11.719354039917562)
(<Say You'll Be There> by MØ, 36.0544217687075, 14.96907911055584)
(<Baby Don't Lie> by Gwen Stefani, 10.204081632653052, 18.253747461523563)
(<Say Something> by Karen Harding, 5.4421768707483045, 20.944380046890736)
(<Sunlight (feat. Years and Years) - Extended Club Mix> by The Magician, 43.537414965986386, 17.834428097570235)
(<Sonnentanz> by Klangkarussell, 40.136054421768705, 10.845772031681467)
(<Blind Heart - Mix Cut> by CAZZETTE, 25.850340136054427, 20.490117402607964)
(<Walk This Way - Kant Remix (Club Version)> by MØ, 26.530612244897956, 19.51170555338354)
(<Embrace> by Goldroom, 0.6802721088435381, 23.145806707645697)
(<Cool Kids - RAC Mix> by Echosmith, 10.884353741496591, 22.481884381386266)
(<Holding Onto Heaven - Chainsmokers Remix> by Foxes, 32.65306122448981, 20.280457720631304)
(<La La La> by Naughty Boy, 10.884353741496591, 21.7131322141385)
(<Ain't Nobody (Loves Me Better) - Extended Mix> by Felix Jaehn, 85.03401360544221, 15.004022390885284)
(<Rihanna (feat. Noonie Bao)> by Clean Bandit, 75.17006802721089, 16.471640164721926)
(<Crying for No Reason> by Katy B, 18.02721088435374, 23.495239510940134)
(<Feelin' Myself> by will.i.am, 74.82993197278913, 17.62476841559357)
(<Ghosttown> by Madonna, 69.421768707483, 18.218804181194116)
(<Sonnentanz - Sun Don't Shine> by Klangkarussell, 47.27891156462585, 18.183860900864673)
(<Uptown Funk> by Mark Ronson, 0.9605122732123808, 7.473431045628695)
(<Dust Clears> by Clean Bandit, 19.423692636072577, 5.8316168200388985)
```

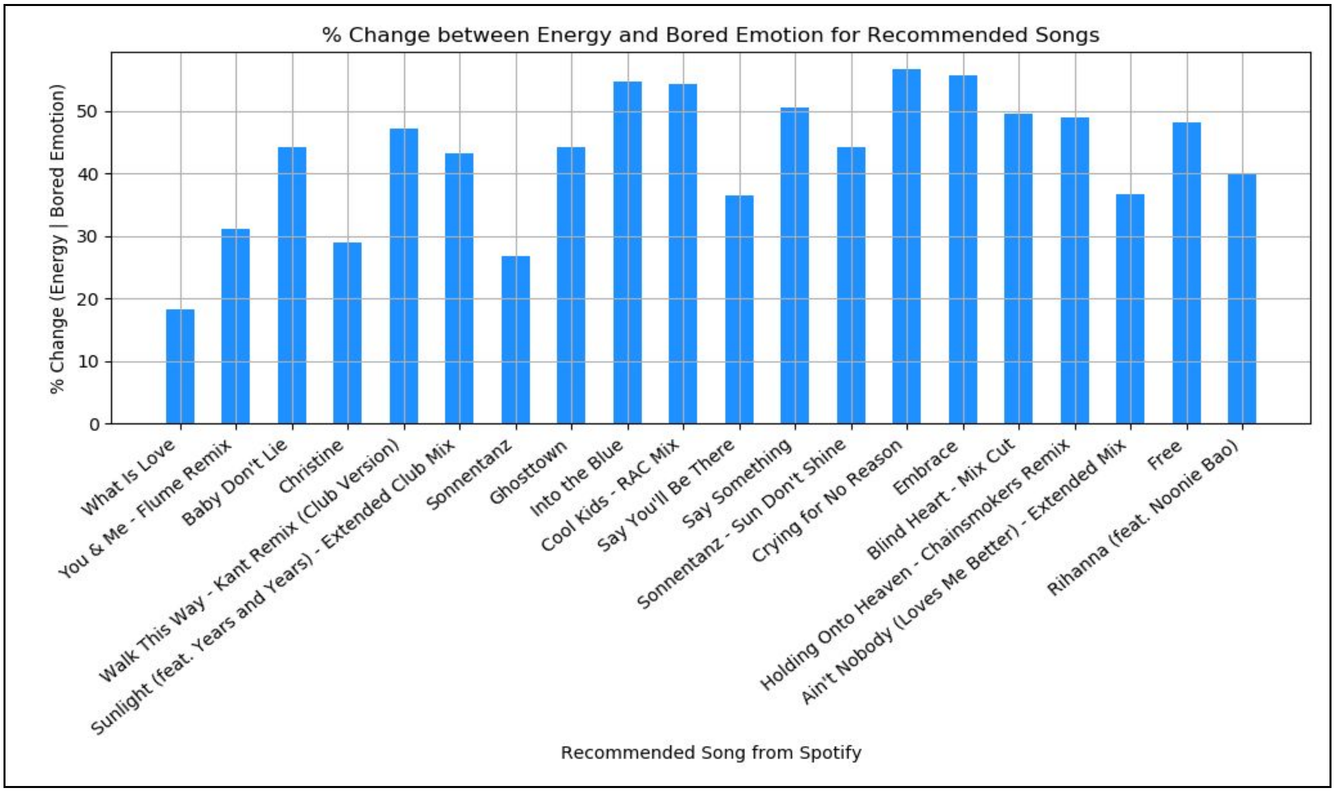
```
(<I Will Never Let You Down - Switch Remix> by Rita Ora, 13.447171824973317, 6.777747051734714)
(<Prayer in C - Robin Schulz Remix> by Lilly Wood and The Prick, 6.296691568836718, 7.65430888404113)
(<Control> by Disclosure, 10.138740661686242, 7.612567844407491)
(<Rumble> by Kelis, 11.953041622198516, 7.306466887094139)
(<Uptight Downtown> by La Roux, 2.668089647812157, 9.115245271218491)
(<Feel Your Love> by Le Youth, 17.502668089647813, 7.2508121675826205)
(<It's My Party> by Jessie J, 6.296691568836718, 8.878712713294536)
(<See Right Through> by Tensnake, 27.001067235859125, 6.4299050547877235)
(<Teach Me> by Bakermat, 26.360725720384213, 6.6246965730780385)
(<Cool Kids> by Echosmith, 16.115261472785487, 8.336079198057233)
(<In Your Eyes (feat. Yandel) - Radio Edit> by INNA, 5.442902881536825, 9.741360865723076)
(<Come Alive (feat. Toro Y Moi)> by Chromeo, 2.5613660618996703, 9.379605188898205)
(<Dance with Me - Extended Mix> by Le Youth, 5.22945570971185, 9.504828307799121)
(<Stranger> by Chris Malinchak, 2.24119530416221, 10.186598621815223)
(<My Love (feat. Jess Glynne)> by Route 94, 22.518676627534692, 7.65430888404113)
(<Come Get It Bae> by Pharrell Williams, 4.482390608324444, 10.103116542547946)
(<Christine> by Christine and the Queens, 51.54749199573106, 4.064579475548185)
(<Watch Out For This (Bumaye)> by Major Lazer, 6.083244397011744, 9.56048302731064)
(<What Is Love> by Kiesza, 5.932203389830502, 6.143582705044595)
(<You & Me - Flume Remix> by Disclosure, 5.932203389830514, 10.916103916499921)
(<Baby Don't Lie> by Gwen Stefani, 11.864406779661028, 15.749421576508817)
(<Christine> by Christine and the Queens, 92.37288135593222, 10.064953636749925)
(<Walk This Way - Kant Remix (Club Version)> by MØ, 8.474576271186438, 16.843757650473094)
(<Sunlight (feat. Years and Years) - Extended Club Mix> by The Magician, 29.661016949152536, 15.384642885187391)
(<Sonnentanz> by Klangkarussell, 25.423728813559322, 9.304998029830289)
(<Ghosttown> by Madonna, 61.906779661016955, 15.719023352232032)
(<Into the Blue> by Kylie Minogue, 8.050847457627114, 19.609996059660578)
(<Cool Kids - RAC Mix> by Echosmith, 11.016949152542383, 19.427606713999864)
(<Say You'll Be There> by MØ, 69.49152542372883, 12.89198849449098)
(<Say Something> by Karen Harding, 31.355932203389838, 18.090084845821302)
(<Sonnentanz - Sun Don't Shine> by Klangkarussell, 34.32203389830509, 15.688625127955246)
(<Crying for No Reason> by Katy B, 2.118644067796612, 20.309155218026643)
(<Embrace> by Goldroom, 23.728813559322035, 20.005172975258787)
(<Blind Heart - Mix Cut> by CAZZETTE, 56.779661016949156, 17.69490793022309)
(<Holding Onto Heaven - Chainsmokers Remix> by Foxes, 65.2542372881356, 17.512518584562375)
(<Ain't Nobody (Loves Me Better) - Extended Mix> by Felix Jaehn, 130.5084745762712, 12.922386718767765)
(<Free> by Rudimental, 32.62711864406779, 17.178138117517733)
(<Rihanna (feat. Noonie Bao)> by Clean Bandit, 118.22033898305087, 14.199112138392755)
(<I Will Never Let You Down - Switch Remix> by Rita Ora, 3.221957040572782, 10.030640255398906)
(<Dust Clears> by Clean Bandit, 9.904534606205248, 8.68880924043088)
(<Rumble> by Kelis, 1.5513126491646791, 10.780486999057507)
```

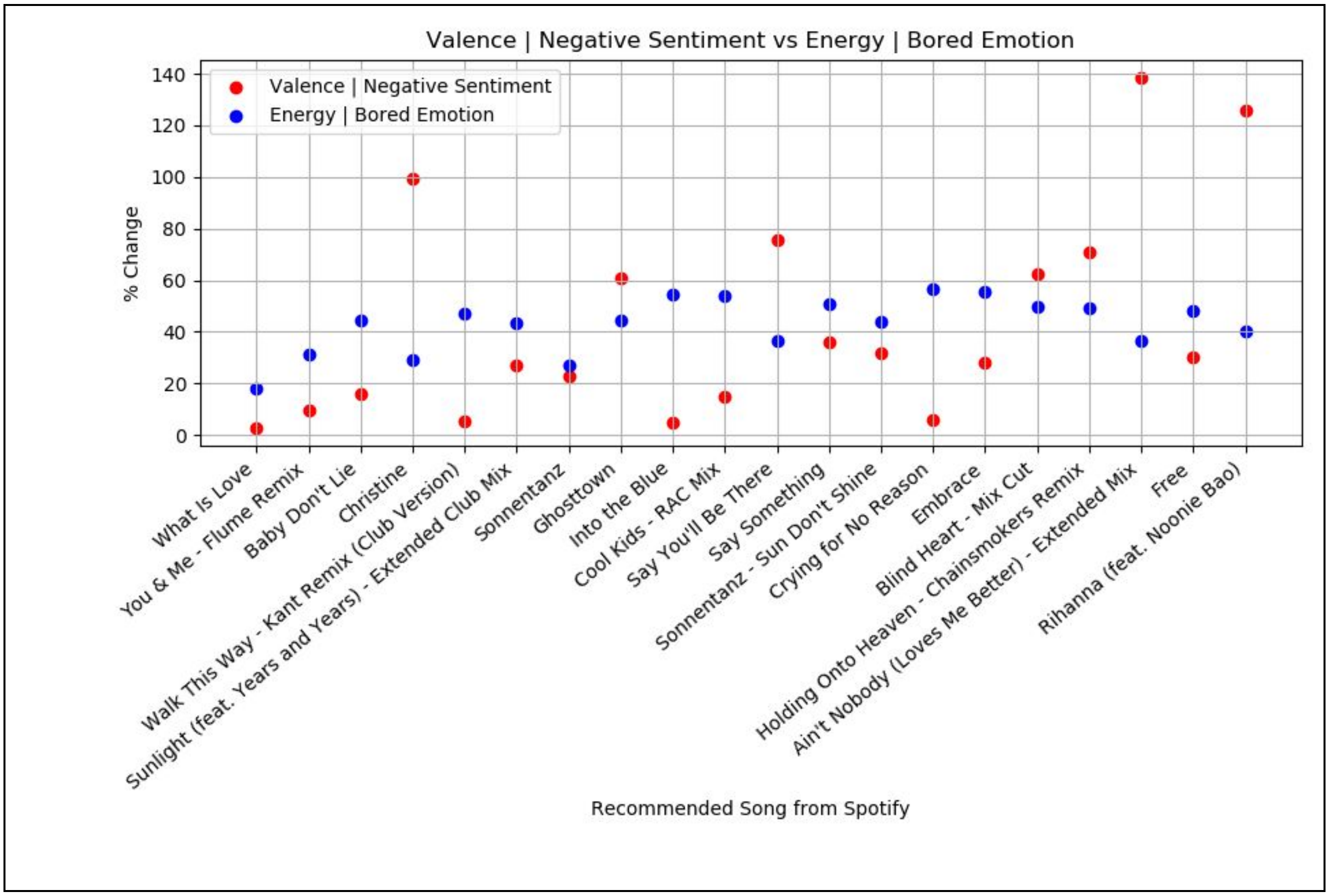
```
(<Control> by Disclosure, 0.47732696897374743, 11.21460879801775)
(<Prayer in C - Robin Schulz Remix> by Lilly Wood and The Prick, 4.773269689737474, 11.273807225148692)
(<See Right Through> by Tensnake, 18.377088305489252, 9.53732002930772)
(<Feel Your Love> by Le Youth, 7.756563245823383, 10.701555762882917)
(<Teach Me> by Bakermat, 17.661097852028643, 9.813579355918783)
(<Uptown Funk> by Mark Ronson, 10.739856801909317, 11.017280707581275)
(<Cool Kids> by Echosmith, 6.205250596658704, 12.240714868287416)
(<Christine> by Christine and the Queens, 45.82338902147971, 6.182742491887659)
(<My Love (feat. Jess Glynne)> by Route 94, 13.365155131264917, 11.273807225148692)
(<Faded> by ZHU, 26.730310262529834, 8.412549913819817)
(<It's My Party> by Jessie J, 4.773269689737474, 13.010294420989664)
(<Come And Get It> by John Newman, 4.415274463007151, 13.168156893338843)
(<Ain't Nobody (Loves Me Better) - Extended Mix> by Felix Jaehn, 35.083532219570394, 8.037626541990516)
(<Something About You> by Hayden James, 18.377088305489252, 10.879151044275742)
(<Bang Bang> by will.i.am, 14.558472553699284, 11.964455541676353)
(<NEW DORP. NEW YORK> by SBTRKT, 34.60620525059665, 7.642970361117568)
(<Right Here, Right Now> by Giorgio Moroder, 11.336515513126487, 12.536707003942128)
```

The percent change between Negative Sentiment and Actual Valence was calculated by utilizing the two database tables Sentiment and Valence with the formula: 100.0 * (ABS(Valence.Valence-Sentiment.NegativeSentiment)/Sentiment.NegativeSentiment)). The percent change between Bored Emotion and Actual Energy was calculated by utilizing the two database tables Emotion and Energy with the formula: 100.0 * (ABS(Energy.Energy-Emotion.BoredEmotion)/Emotion.BoredEmotion)).

Visualization







Instructions for running code

Instructions to produce an output file of data and a database file

1. Open the "emotify.py" file
2. Get the authorization token for the Spotify API Recommendations. Go to https://developer.spotify.com/documentation/web-api/reference/browse/get-recommendations/ and scroll to the Examples section. Click on the green "Try it" button. Scroll to the bottom of the page, then click on the green "Get Token" button and authorize Spotify access. Copy and paste this OAuth token to the recommendationToken variable as a string. The token expires after one hour, so you will need to click on the "Get Token" button again and copy and paste the new oAuth token.

3. Get the authorization token for the Spotify API Audio Features. Go to https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/ and scroll down to the green "Try It" button. Click on the green "Get Token" button and authorize Spotify access. Copy and paste this OAuth token to the featureToken variable as a string. The token expires after one hour.
4. (Optional) Alter the content of the list of strings in the main method for the variable called "input_text" to get different results. The strings of text are what determine the calculated levels of bored emotion and negative sentiment through the ParallelDots analysis, which then set the target levels of valence and energy for recommended songs from Spotify.
5. Run the emotify.py file to produce a database and output file of data.

Instructions to produce visualizations

1. Make sure the file "emotifyoutput.txt" exists (it would have been created after running the file "emotify.py") and is passed into all of the visualization files.
2. Run "visualization1.py" to produce a bar graph which displays the percent change between the values for the actual valence for the recommended song and the negative sentiment of the text.
3. Run "visualization2.py" to produce a bar graph which displays the percent change between the values for the actual energy for the recommended song and the bored emotion value of the text.
4. Run "visualization3.py" to produce a scatterplot which plots the percent changes between the percent changes between actual valence/negative sentiment values and actual energy/bored emotion values.

Documentation for each method

emotify.py

- ***setUpDatabase*** creates a database file named with the string passed into db_name.
- ***setUpSpotifyValence*** inputs the API access token as "featureToken", the Spotify recommendation data as "data", the database cursor as "cur", and the database connection

object as “conn”. This then outputs the “SpotifyID” as text for the Spotify ID of a recommended song, “Song” as text for the title of the recommended song, “Artist” as text for the name of artist, and “Valence” as float for the valence level of the recommended song into the database table called “Valence”.

- ***setUpSpotifyEnergy*** inputs the API access token as “featureToken”, the Spotify recommendation data as “data”, the database cursor as “cur”, and the database connection object as “conn”. This then outputs the “SpotifyID” as text for the Spotify ID of a recommended song, “Song” as text for the title of the recommended song, “Artist” as text for the name of artist, and “Energy” as float for the energy level of the recommended song into the database table called “Energy”.
- ***setUpSentiment*** inputs the calculated sentiment data from the ParallelDots API as “sentiment_data”, Spotify recommendation data from the Spotify API as “spotify_data”, the database cursor as “cur”, and the database connection object as “conn”. This then outputs the “Count” as an integer for indicating the row number in the database (how many rows of data), “TextID” as an integer to indicate which index of the text string that the data is for, and “NegativeSentiment” as a float to indicate the negative sentiment score for a text string into the database table Sentiment.
- ***setUpEmotion*** inputs the calculated sentiment data from the ParallelDots API as “emotion_data”, Spotify recommendation data from the Spotify API as “spotify_data”, the database cursor as “cur”, and the database connection object as “conn”. This then outputs the “Count” as an integer for indicating the row number in the database (how many rows of data), “TextID” as an integer to indicate which index of the text string that the data is for, and “BoredEmotion” as a float to indicate the bored emotion score for a text string into the database table Emotion.
- ***setUpEmotify*** passes in a database cursor as “cur” and a database connection object as “conn” in order to select data from all of the tables in a JOIN statement, while also calculating the percent change between the negative sentiment and actual valence values along with the bored emotion and actual energy values for each song. The values are then written to a text file with the recommended song titles, artists, and calculated values.
- ***main*** creates a database file called “emotify.db” and then takes the gathers data using the Spotify and ParallelDots APIs. ParallelDots evaluates the list of strings in ‘input_text’ and

outputs the data into the empty lists ‘negative_valence’ and ‘bored_energy’. The Spotify API uses these float values as parameters for the target valence and target energy calculated in this method in order to generate the recommended songs’ data which are then passed into the “setUpSpotifyValence” and “setUpSpotifyEnergy” methods to create the tables for Valence and Energy in the database file. This method also executes the “setUpSentiment” and “setUpEmotion” methods to pass the Spotify data and create tables for Sentiment and Emotion. The calculated data in “setUpEmotify” from selected values in the database is executed and outputted into the text file. Finally, the database connection is closed at the end.

visualization1.py

- Reads the “emotifyoutput.txt” file line by line, stripping unnecessary characters such as “\n” and appends the values to a list of recommended songs and a list of percent changes percent change between the values for the actual valence for the recommended song and the negative sentiment value of the text.
- Sets up the bar graph with axes and plots the values from the lists.
- Output is a bar graph which displays the percent change between the values for the actual valence for the recommended song and the negative sentiment of the text.

visualization2.py

- Reads the “emotifyoutput.txt” file line by line, stripping unnecessary characters such as “\n” and appends the values to a list of recommended songs and a list of percent changes between the values for the actual energy for the recommended song and the emotion value of the text.
- Sets up the bar graph with axes and plots the values from the lists.
- Output is a bar graph which displays the percent change between the values for the actual energy for the recommended song and the actual emotion of the text.

visualization3.py

- Reads the “emotifyoutput.txt” file line by line, stripping unnecessary characters such as “\n” and appends the values to a list of recommended songs, a list of percent changes between actual valence and negative sentiment values, then a list of percent changes between actual energy and bored emotion values.
- Sets up the scatter plot and plots the values from the lists.
- Output is a scatter plot which plots the percent changes between the percent changes between actual valence/negative sentiment values and actual energy/bored emotion values.

Resources

Date	**Issue Description**	**Location of Resource**	**Result**
12/9/19	Converting cURL syntax to python	https://curl.trillworks.com/	Successfully converted the cURL syntax from the Spotify API documentation to Python (for both Spotify tables in our database)
12/10/19	Missing code for getting song recommendations from Spotify	https://developer.spotify.com/documentation/web-api/reference/browse/get-recommendations/	This code allowed for us to get song recommendations based on features and seed tracks
12/10/19	Missing code to get audio features of each	https://developer.spotify.com/documenta	This code allowed for us to see the audio

	song from Spotify	tion/web-api/reference/tracks/get-audio-features/	features of each song that was in the recommendations table in the database
12/10/19	Missing code to get the level for each emotion from ParallelDots	http://apis.paralleldots.com/text_docs/index.html#emotion	This code allowed for us to see the level of each emotion for each string in our list of strings
12/10/19	Missing code to get the level for each sentiment from ParallelDots	http://apis.paralleldots.com/text_docs/index.html#sentiment	This code allowed for us to see the level of sentiment (positive, negative, neutral) for each string in our list of strings
12/11/19 - 12/17/19	TypeError, NameError, SyntaxError	https://stackoverflow.com/	Helped with debugging and solving errors, particularly with syntax. Also was helpful in determining how to calculate data using SQL.
12/11/19-12/17/19	Needed to know how SQLite-specific syntax worked and how to fully implement	https://www.sqlite.org/docs.html	Helped with creating the databases and calculating data selected from all the

	features of SQL to create the database		tables in a join
12/16/19	Needed to be able to select data from database and calculate values from it	https://www.w3schools.com/sql/func_sqlserver_abs.asp,	Was able to calculate the percent change between values in the database file using absolute value
12/16/19	Needed to select columns from multiple tables and join them together to calculate data	https://academy.vertabelo.com/blog/illustrated-guide-multiple-join/	Was able to select columns and join them together in order to calculate data
12/16/19	Needed to calculate the percent change between data values in the database (was unsure about SQL math syntax)	http://doc.nuodb.com/Latest/Content/SQL-Mathematical-Functions-and-Operators.htm	Was able to take selected data from the database and calculate the percent change between values
12/17/19	Needed to visualize data from our output file as a bar graph and scatter plot	https://matplotlib.org/gallery/index.html	Created bar graphs and a scatter plot using our data from "emotifyoutput.txt"
12/17/19	Needed to write data to an output file	https://www.w3schools.com/python/python_file_write.asp	Was able to write data to file using selected database values