John Espinoza, Tom Stowell, Choua Vang, Juniper Cherne, Zach Diercks

Capstone Project, Group 8

Final Written Analysis

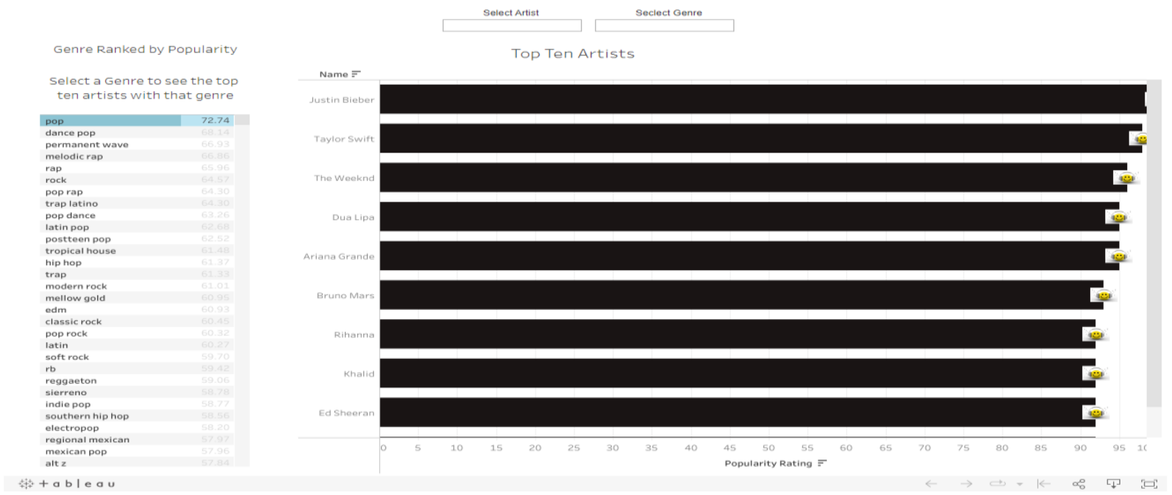
Introduction

The goal of our project is to create a recommender model that will allow a user to get song recommendations they may enjoy. We are utilizing the Kaggle Spotify Dataset sampling over 600,00 songs to build our model from. Our inspiration came from our love of music and that regardless of anyone's demographics can listen to their favorite songs and artists. Additional inspiration came from reviewing the Book Recommender that was completed in a previous bootcamp. Our expectation from our model is to provide 10 songs that are similar to the user's input. This recommender model is based on multiple song characteristics within our dataset.

Tableau

We created two dashboards for our website. The first focuses on the artist and the genre that they are classified into. The second focuses on the artist tracks, genre, and musical attributes. Which we then tracked over time.

Our 1st dashboard ranks music by popularity. We have two visualizations on this dashboard. Our first chart ranks the genre by popularity. The second ranks the artists by popularity. You can hover over the "Top Ten Artists" chart, and it will also display all of the genres of that particular artist.



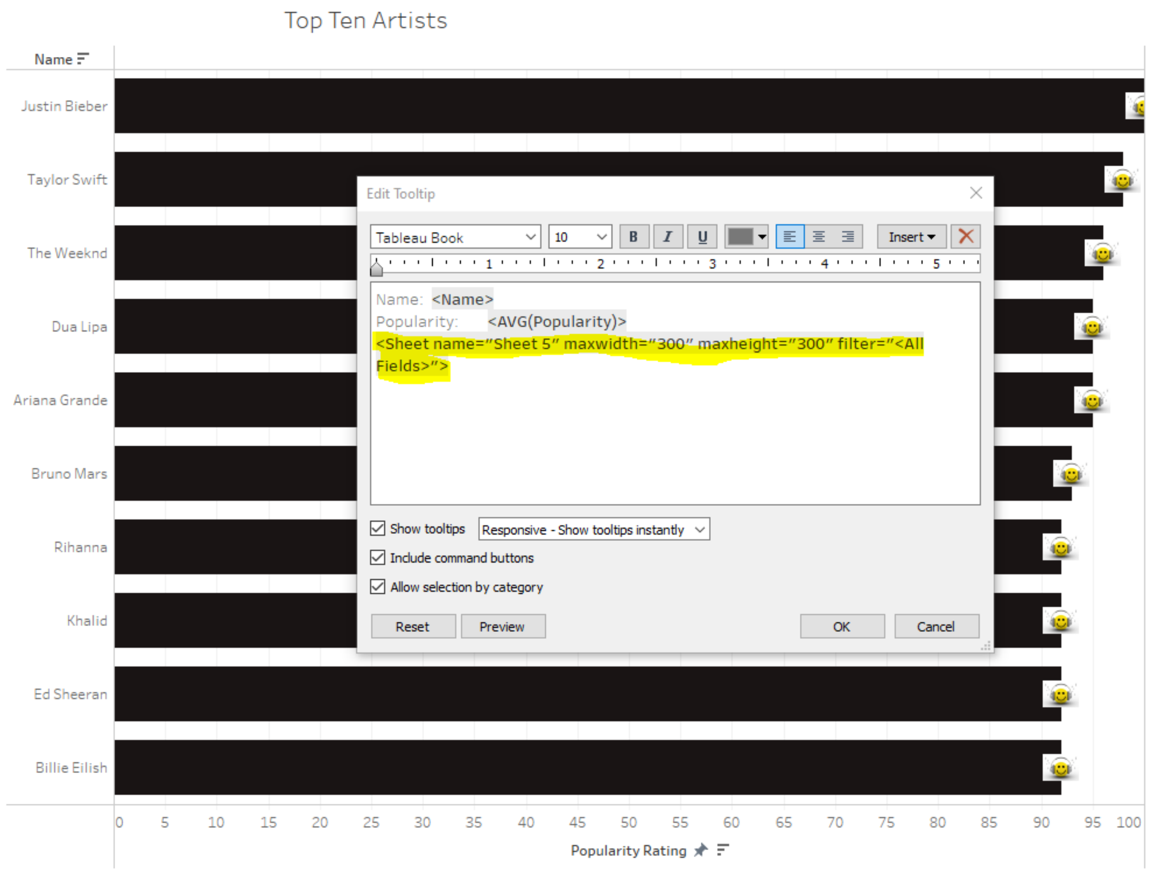
Genre Ranked by Popularity

Select a Genre to see the top ten artists with that genre

pop	72.74
dance pop	68.14
permanent wave	66.93
melodic rap	66.86
rap	65.96
rock	64.57
pop rap	64.30
trap latino	64.30
pop dance	63.26
latin pop	62.68
postteen pop	62.52
tropical house	61.48
hip hop	61.37
trap	61.33
modern rock	61.01
mellow gold	60.95
edm	60.93
classic rock	60.45
pop rock	60.32
latin	60.27
soft rock	59.70
rb	59.42
reggaeton	59.06
sierreno	58.78
indie pop	58.77
southern hip hop	58.56
electropop	58.20
regional mexican	57.97
mexican pop	57.96
alt z	57.84

The dashboard can be updated by can selecting one of the genres from the "genre ranked by popularity" chart, this will then update the "Top Ten Artists" chart, with artists from that genre. The second way to update the dashboard is to type in artist or genre in the selection boxes at the top of the dashboard. This will update both charts to pull only information related to your selection in the selection box. This feature is useful if you know an artist or genre that you want to quickly see the results for.

Challenges for building the first dashboard resolved around creating the pop-up window to display all the genres that an artist may be belong to in the "Top Ten Artists" chart. The genre data field had multiple genres for the artist in one field. We overcame this limitation by using tableau prep and using the pivot table option to parse out the genre for each artist into their own row of data. Next, we could not get the genres to display correctly within the pop-up window. This was resolved by creating a new worksheet with just the genres as a matrix and then using the tooltip function within our bar chart and inserting the new worksheet as the data. The tooltip insert step can be seen below:

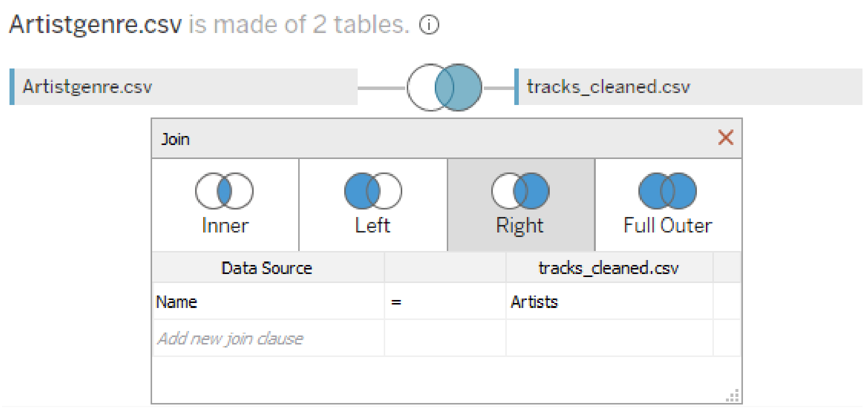


Name: <Name>
Popularity: <AVG(Popularity)>
<Sheet name="Sheet 5" maxwidth="300" maxheight="300" filter="<All Fields>">

☑ Show tooltips — Responsive - Show tooltips instantly
☑ Include command buttons
☑ Allow selection by category

The 2nd dashboard explores seven audio analysis features of each song and visualizes how they have transformed throughout time with the ability to utilize global filters on specific or multiple genres and decades. An additional visual identifies what were the most popular genres each decade. All visuals within the dashboard respond to any change in the genre or decade filters, whether one, multiple or all but one.

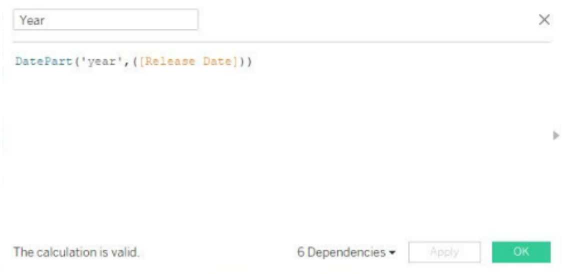
The first step in reviewing the data was creating a Right-Join between our two data sets. This would allow us to tie the release date of each song to the multiple genres of the artist.


Artistgenre.csv is made of 2 tables.

Artistgenre.csv — tracks_cleaned.csv

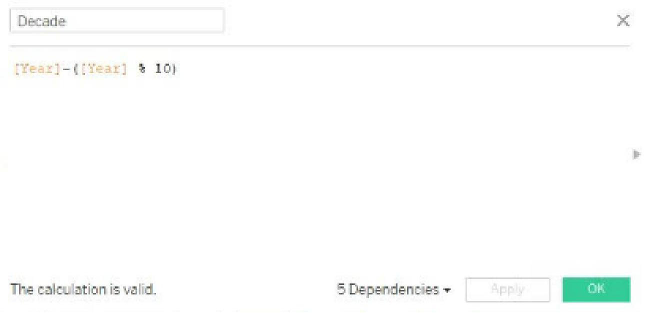
Data Source		tracks_cleaned.csv
Name	=	Artists
Add new join clause		

After the join was completed, I worked on creating two calculated fields to identify each songs release date, "Year" and then generated a decade set to utilize as a global filter.


Year

```
DatePart('year',([Release Date]))
```

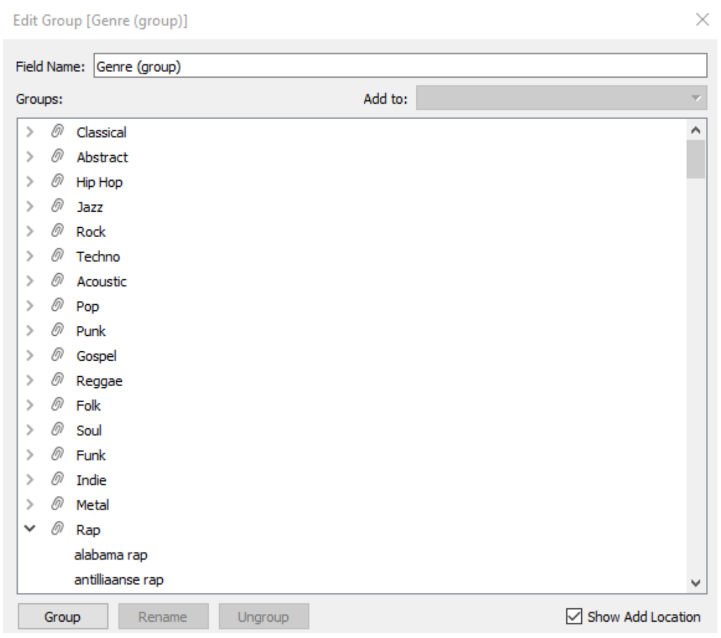
The calculation is valid.


Decade

```
[Year]-([Year] % 10)
```

The calculation is valid.

Due to the large number of various genres within our data set I created a Genre (group) based on similar genre names to narrow down the genre search within the global filters. For example, this would allow the user to easily select all various "country" genres with one click to review each visual.


Edit Group [Genre (group)]

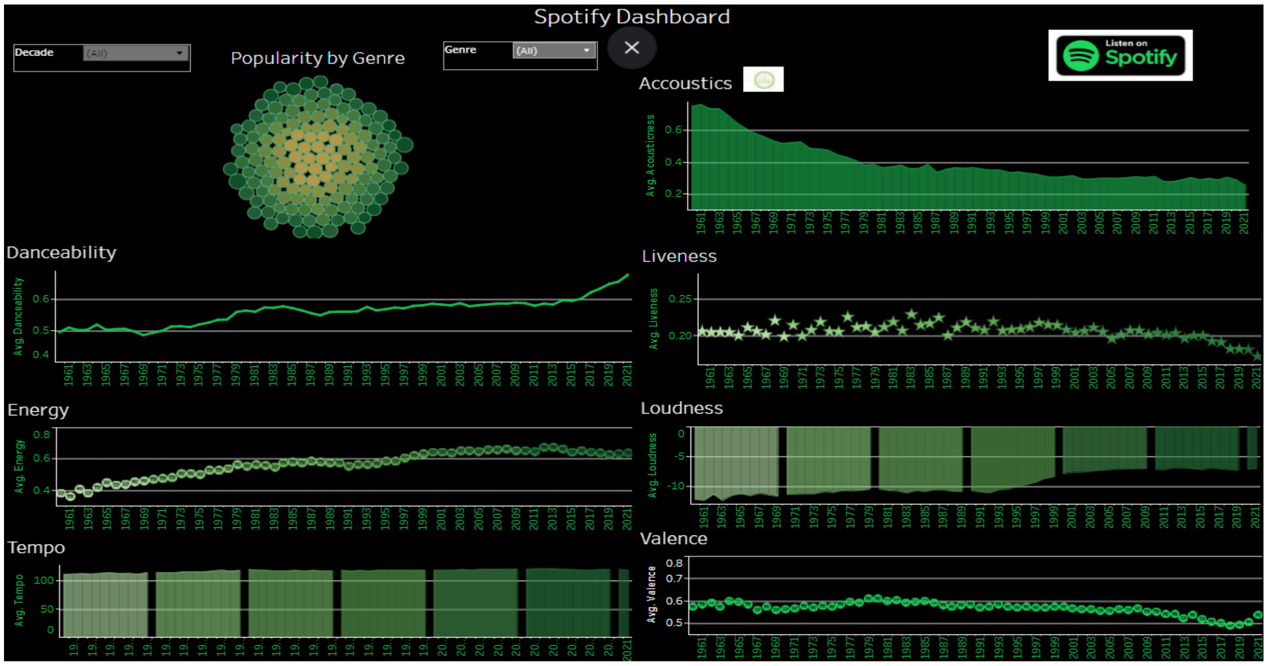
Field Name: Genre (group)

Groups: Add to:

- Classical
- Abstract
- Hip Hop
- Jazz
- Rock
- Techno
- Acoustic
- Pop
- Punk
- Gospel
- Reggae
- Folk
- Soul
- Funk
- Indie
- Metal
- Rap
  - alabama rap
  - antilliaanse rap

Group | Rename | Ungroup | ☑ Show Add Location

The data compiled is averaging each metric based on genre and audio feature for each year and decade. The popularity of genre takes the average of each genre for the entire decade to see the transformation over time. The data shows that danceability, energy, loudness continue to trend up the past sixty years, while acoustics and liveness have trended down. Tempo and valence have remained steady over the past sixty years.



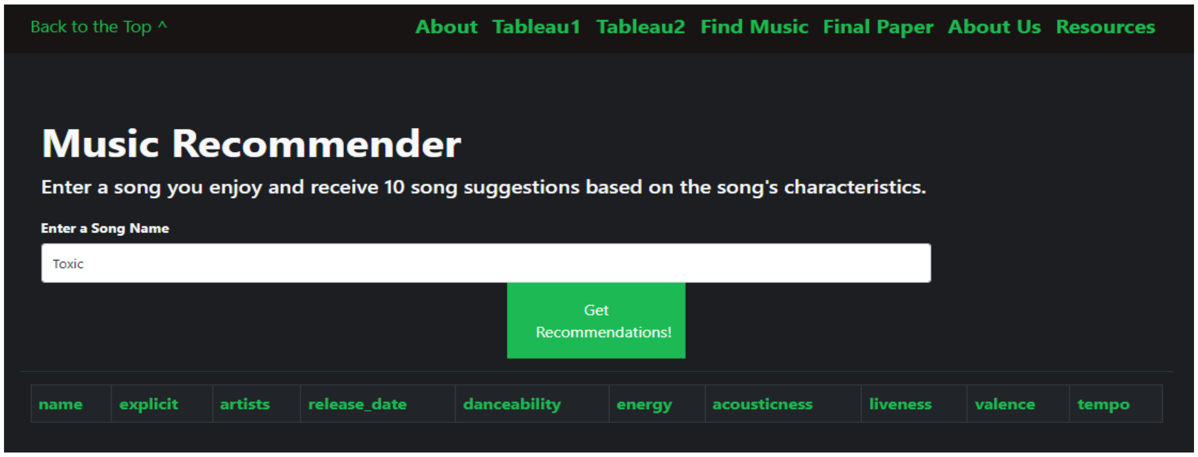
Additional attributes to this dashboard include the color scheme, information button, and link to Spotify. The color scheme is based on the official Spotify colors (Hex Color: #1DB954 & #191414). The information button identifies to the user how Spotify distinguishes each audio metric for the user to reference. The Spotify logo will bring you to the login page the Spotify webpage. Another attribute was updating the shape logo to be the Spotify logo vs a standard Tableau shape option.

Difficulties that we experienced when reviewing the data and creating these dashboards were that many songs and artists had multiple genres assigned to them. We utilized Tableau Prep to further clean the data to further improve our data when creating these dashboards. Another difficulty was the sheer number of genres within our dataset but were able to manually group these within Tableau.

Both dashboards were embedded and hosted on Tableau Public.

<u>Web page</u>

Our webpage design was selected from StartBootstrap.com templates. Using CSS stylesheets and HTML we made customizations to the template. This template was selected not only for its simple look but also for its easy-to-use functions that allows for continued user engagement throughout the page. The navigation bar is easily accessible and utilized throughout the user's time on our page. It includes buttons to navigate the user to specific sections and back to the top of the page.



Music Recommender

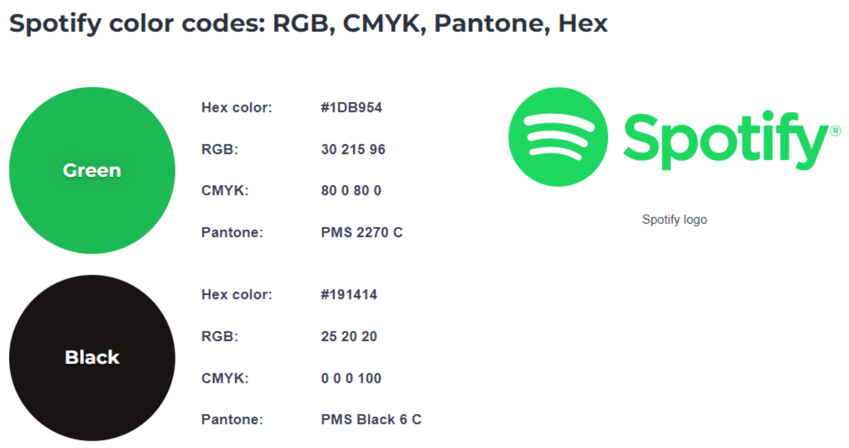
Enter a song you enjoy and receive 10 song suggestions based on the song's characteristics.

Enter a Song Name

Toxic

name	explicit	artists	release_date	danceability	energy	acousticness	liveness	valence	tempo

On our page you can view two tableau dashboards, our music recommender, information about us, final paper and works cited. The theme was inspired by the Spotify colors and dataset used. The color palette chosen is from U.S. Brand Colors:

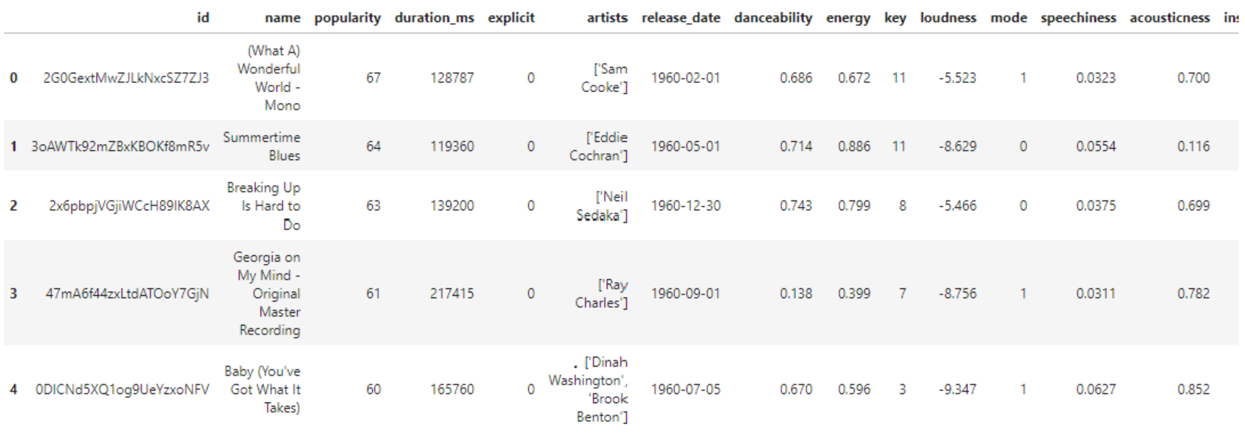


Spotify color codes: RGB, CMYK, Pantone, Hex

Green	Hex color:	#1DB954
	RGB:	30 215 96
	CMYK:	80 0 80 0
	Pantone:	PMS 2270 C
Black	Hex color:	#191414
	RGB:	25 20 20
	CMYK:	0 0 0 100
	Pantone:	PMS Black 6 C

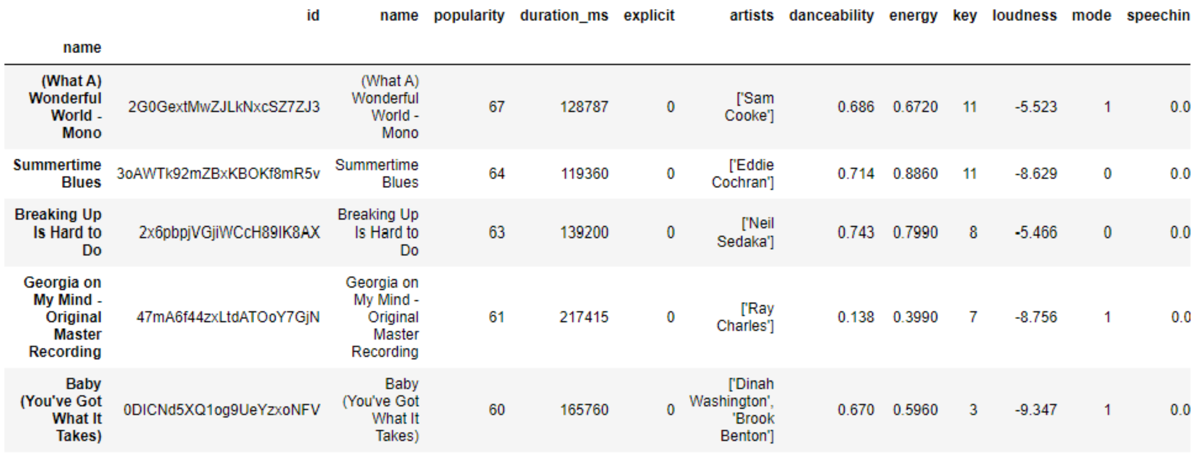
Pre-Processing

The process starts by reading in the cleaned csv file that contains the information regarding the song and its metrics that was prepared by in the data cleaning steps. The metrics are as follows: id, name, popularity duration_ms, explicit, artists, release_date, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, and tempo.



	id	name	popularity	duration_ms	explicit	artists	release_date	danceability	energy	key	loudness	mode	speechiness	acousticness	in
0	2G0GextMwZJLkNxcSZ7ZJ3	(What A) Wonderful World - Mono	67	128787	0	['Sam Cooke']	1960-02-01	0.686	0.672	11	-5.523	1	0.0323	0.700	
1	3oAWTk92mZBxKBOKf8mR5v	Summertime Blues	64	119360	0	['Eddie Cochran']	1960-05-01	0.714	0.886	11	-8.629	0	0.0554	0.116	
2	2x6pbpjVGjiWCcH89IK8AX	Breaking Up Is Hard to Do	63	139200	0	['Neil Sedaka']	1960-12-30	0.743	0.799	8	-5.466	0	0.0375	0.699	
3	47mA6f44zxLtdATOoY7GjN	Georgia on My Mind - Original Master Recording	61	217415	0	['Ray Charles']	1960-09-01	0.138	0.399	7	-8.756	1	0.0311	0.782	
4	0DICNd5XQ1og9UeYzxoNFV	Baby (You've Got What It Takes)	60	165760	0	. ['Dinah Washington', 'Brook Benton']	1960-07-05	0.670	0.596	3	-9.347	1	0.0627	0.852	

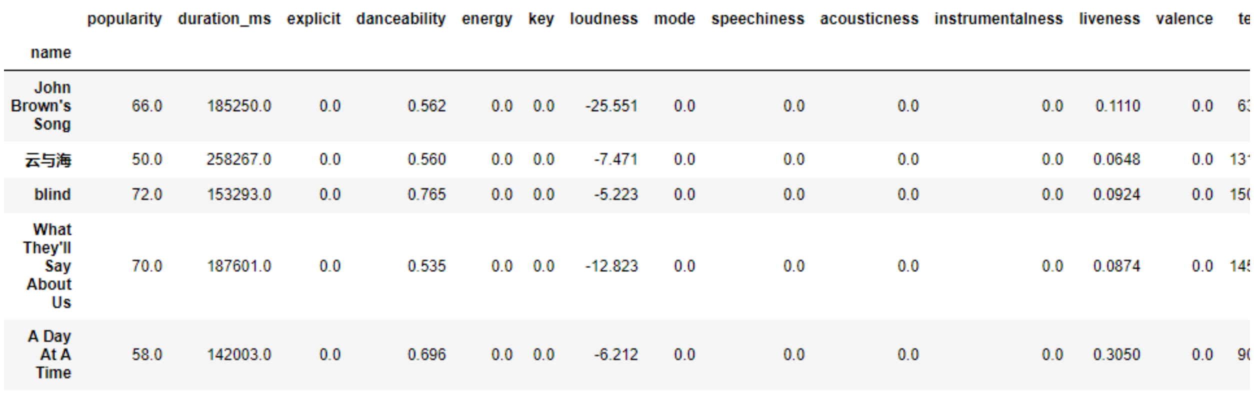
This import takes place twice and one is saved as KNN_top_tracks_alpha and the other is saved as KNN_top_tracks. Duplicate songs in the name column are dropped in both data frames. In the KNN_top_tracks_alpha data frame the index is reset but left as a numeric as the recommender will iterate through the tracks and return a song with its corresponding index number. In the KNN_top_tracks alpha the index is set to the name of the song and the id column is dropped.



	id	name	popularity	duration_ms	explicit	artists	danceability	energy	key	loudness	mode	speechin
name												
(What A) Wonderful World - Mono	2G0GextMwZJLkNxcSZ7ZJ3	(What A) Wonderful World - Mono	67	128787	0	['Sam Cooke']	0.686	0.6720	11	-5.523	1	0.0
Summertime Blues	3oAWTk92mZBxKBOKf8mR5v	Summertime Blues	64	119360	0	['Eddie Cochran']	0.714	0.8860	11	-8.629	0	0.0
Breaking Up Is Hard to Do	2x6pbpjVGjiWCcH89IK8AX	Breaking Up Is Hard to Do	63	139200	0	['Neil Sedaka']	0.743	0.7990	8	-5.466	0	0.0
Georgia on My Mind - Original Master Recording	47mA6f44zxLtdATOoY7GjN	Georgia on My Mind - Original Master Recording	61	217415	0	['Ray Charles']	0.138	0.3990	7	-8.756	1	0.0
Baby (You've Got What It Takes)	0DICNd5XQ1og9UeYzxoNFV	Baby (You've Got What It Takes)	60	165760	0	['Dinah Washington', 'Brook Benton']	0.670	0.5960	3	-9.347	1	0.0

This was done during the first pass through of making the recommender when the index number was directly referenced but given the alteration of the recommender code it is quite possible that a single data frame could be utilized. In possible updates to the program, especially its preprocessing, this can be resolved.

From here the KNN_top_tracks_proc data frame is created by dropping the 'release date' column from the KNN_top_tracks data frame. This is because this column will not be used in the Nearest Neighbors algorithm to determine like songs due to it not being a float numerical data point. To make data that can be utilized by the Nearest Neighbors algorithm, the remaining metrics, 'popularity' 'duration_mn', 'explicit', 'energy', 'key', 'mode', 'speechiness', 'acousticeness', 'instrumentalness', and 'valence,' and transformed in to the 'float' format. Initially when performing the preprocessing the Standard Scaler was then used on these values, however, after evaluation this step was not included because the data was already scaled from 1 to 100 on the majority of the scored categories. After this the 'name', 'artist', and 'id' columns are dropped as they too will not be utilized during the running of the Nearest Neighbors algorithm that takes place in the song recommender function.



	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	te
name														
John Brown's Song	66.0	185250.0	0.0	0.562	0.0	0.0	-25.551	0.0	0.0	0.0	0.0	0.1110	0.0	6
云与海	50.0	258267.0	0.0	0.560	0.0	0.0	-7.471	0.0	0.0	0.0	0.0	0.0648	0.0	13
blind	72.0	153293.0	0.0	0.765	0.0	0.0	-5.223	0.0	0.0	0.0	0.0	0.0924	0.0	15
What They'll Say About Us	70.0	187601.0	0.0	0.535	0.0	0.0	-12.823	0.0	0.0	0.0	0.0	0.0874	0.0	14
A Day At A Time	58.0	142003.0	0.0	0.696	0.0	0.0	-6.212	0.0	0.0	0.0	0.0	0.3050	0.0	9

These data frames are then exported as CSV files to be utilized in conjunction with the app.py file in the song recommender on the web page.

Song Recommender

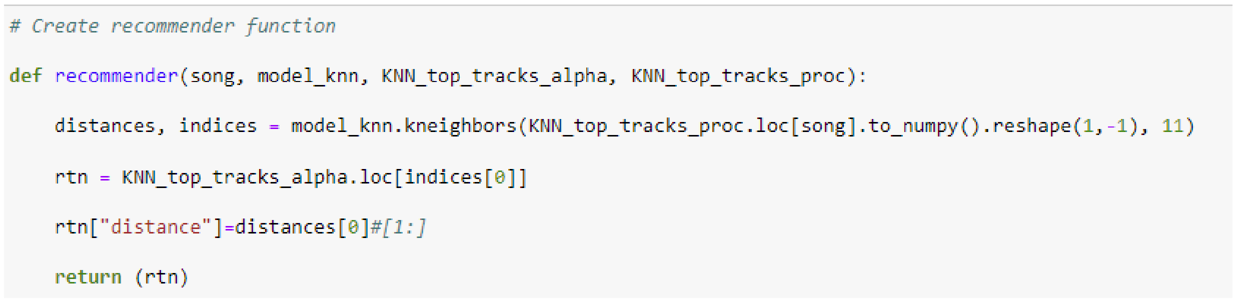
The algorithm utilized for the song recommender is the Nearest Neighbors method form scikit-learn. The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point and predict that label from these. When establishing the knn model you select the metric, algorithm, and number of n_neighbors to use.

```
# Establish model using the NearestNeighbors algorithm

model_knn = NearestNeighbors(metric='cosine', algorithm = 'brute', n_neighbors = 10)
```

From there we fit the KNN_top_tracks_proc data frame with the generated knn model.

```
# Fit the KNN_top_tracks_proc with the model

model_knn.fit(KNN_top_tracks_proc)
```

It is after this that these are used to develop the recommender function as show below.



```
# Create recommender function

def recommender(song, model_knn, KNN_top_tracks_alpha, KNN_top_tracks_proc):

    distances, indices = model_knn.kneighbors(KNN_top_tracks_proc.loc[song].to_numpy().reshape(1,-1), 11)

    rtn = KNN_top_tracks_alpha.loc[indices[0]]

    rtn["distance"]=distances[0]#[1:]

    return (rtn)
```

The recommender code shows that there four inputs into the function. They are the input ‘song’, the knn model ‘model_knn’, and the data frames KNN_top_tracks_alpha and KNN_top_tracks_proc. In this function the distances and indices between and for the data points of the songs based on their similar attributes are extracted. After this, two different values are returned. These are the songs from the KNN_top_tracks_alpha based on their indices and their distances. When the function is ran with the input of the song the ten most similar songs, along with the selected song, are returned because in the model the n_neighbors value was 11.

The recommend function took some time to refine but once completed it is functional and works smoothly. It iterates through nearly 500,000 songs to find those that are most similar to the one input based on the metrics provided. This results in suggestions that an individual may have not been exposed to normally, but they may ultimately enjoy.

<u>Conclusion</u>

Overall, the final product of our project was a major success. We created a machine learning recommender model that we feel users can rely on to provide song recommendations similar to their user input. The machine learning recommender model along with our tableau dashboards and designed web app not only provides song recommendations for the user, but also tells a story of who the most popular artists are based on Spotify records and tells a story of how music has changed over the last sixty years. The user is able to input their song selection and interact with both dashboards to learn more about the attributes that contribute to the recommender model.

Call to Action

Based on our analysis, we came to the following conclusions:

- Danceability, Energy, and loudness of songs continue to increase over the past sixty years
- While Acoustics, Liveness, and Valence have decreased
- Tempo has remained steady over the past sixty years
- Popularity of each specific genre have changed slightly decade by decade
    - Callouts
        - Country and Rap genre's popularity has continued to increase every decade since 1960
        - Metal and Rock genre's popularity has continued to decrease every decade since 1960

Take our recommendations based off our recommender model and go to your preferred music streaming service to listen to the songs provided in our list.

Limitations and Future Work

Within our project and after analyzing our data we found a few limitations that if we had additional time and resources could have improved our analysis, recommender models and web page. One limitation was missing the region and language of the artist that would have been beneficial to the user to provide music within the same region they were looking for. Our data set had music samples from all over the world, but without a region and language category made this impossible to determine without adding another dataset within our analysis. With this additional data we could complete further analysis as well regarding genre's and how they compare in popularity and changed over time by region.

Another limitation was that was we only had popularity of the artist and not the individual track itself. This would have been a beneficial addition as well to the dataset as some artists were a part of multiple genre's that could skew our data when looking at popularity of each artist and genre.

Future Work

In the future we would like to create a predictive text feature with our recommender text box to search for the song that the user is entering into the text field. This would allow the user to only type a portion of the song title and provide options to select. Also, if we had more time, we would have liked to create an additional recommender model to search by artist and return similar artists and song options for the user to explore. The last feature that would have been nice to add within the web page was to have each song suggested with a link to Spotify to play instantaneously. This would eliminate the user needing to search withing Spotify to find the correct artist and song.

Works Cited

- Spotify Audio Analysis Resources
  - https://medium.com/@boplantinga/what-do-spotifys-audio-features-tell-us-about-this-year-s-eurovision-song-contest-66ad188e112a
  - https://developer.spotify.com/discover/

- Tableau, Add Info Button
  - https://www.youtube.com/watch?v=dkV8HrvMr5A

- Adding images and links in Tableau
  - https://www.google.com/search?q=add+logo+to+tableau+dashboard&rlz=1C1VDKB_enUS1015US1015&oq=insert+logo+to+tablea&aqs=chrome.1.69i57j0i22i30l3j0i390l2.7840j0j7&sourceid=chrome&ie=UTF-8
  -
- Removing null Characters in Tableau dataset
  - https://tarsolutions.co.uk/blog/tableau-hide-values-quick-filter/

- Spotify
  - https://open.spotify.com/

- Spotify Color Scheme
  - https://usbrandcolors.com/spotify-colors/

- HTML Template
  - Scrolling Nav - One Page Scrolling Bootstrap Template - Start Bootstrap

- Background Image
  - https://wallpapersafari.com/spotify-backgrounds/

- Links with images
  - HTML Images (w3schools.com)

- KNN Recommender Research sites
  - https://realpython.com/knn-python/
  - https://www.kaggle.com/code/amolbhivarkar/knn-for-classification-using-scikit-learn/notebook
  - https://www.kaggle.com/code/prashant111/knn-classifier-tutorial
  - https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea
  - https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-part-2-k-nearest-neighbors-and-matrix-c04b3c2ef55c
  - https://beckernick.github.io/music_recommender/
  - https://towardsdatascience.com/spotify-genre-classification-algorithm-88051db23d42
  - https://www.kaggle.com/code/vatsalmavani/music-recommendation-system-using-spotify-dataset
  - https://github.com/raj4tshenoy/Music-Recommender-Engine-KNN/blob/master/music_recommender_engine_knn.ipynb